王掌柜

04月09日

第三章 | IFTTT 的本地应用

| 本文为付费栏目文章，您已订阅，可阅读全文 |

本篇属于「玩转 IFTTT 系列」的第三章本地应用篇，如果你对 IFTTT 基础不了解，或者还没有制作过 IFTTT 自动化任务的同学，建议按照顺序阅读本系列的前两篇《初识篇》和《基础篇》，我会带着大家由浅入深的学习并掌握 IFTTT。

在上一篇文章中我们通过实际动手操作，了解了如何通过 IFTTT 创建「自动化任务」。

创建任务的前提是手机上安装了 IFTTT 的客户端，通过客户端才可以创建自动化任务，让 IFTTT 为我们服务。那么除了创建任务之外，IFTTT 的应用还有什么用处?

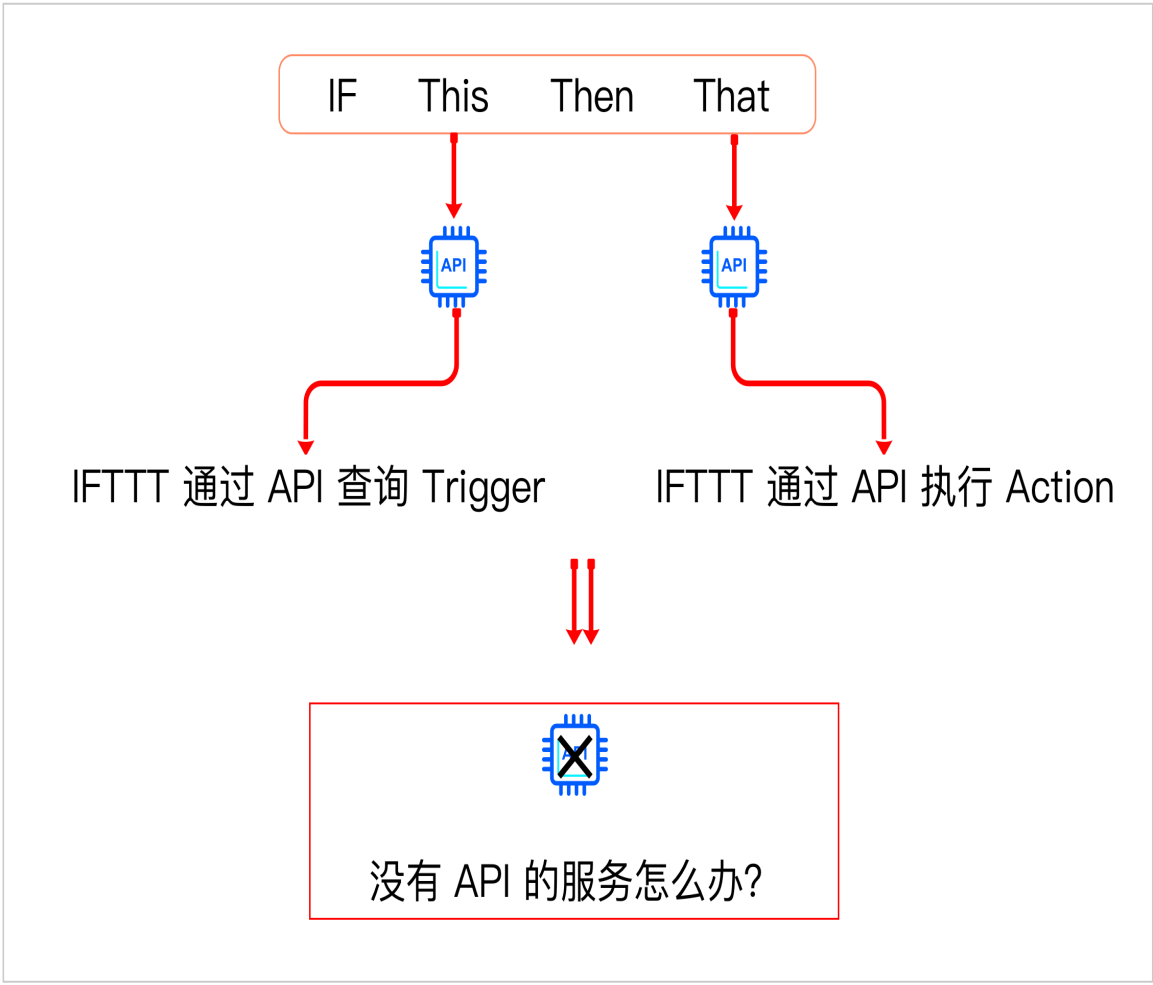
关于这个问题，总的来说，IFTTT 客户端除了创建自动化任务和接收服务通知

关于这个问题，总的来说，IFTTT 客户端除了创建自动化任务和接收服务通知外还具备以下作用：

1. 使没有 API 的本地应用可以通过 IFTTT 实现自动化；
2. 通过 Widget[1]，手动触发 IFTTT 使其完成自动化任务。

接下来我们逐一解释一下这两个作用。

01. 使没有 API 的本地应用可以通过 IFTTT 实现自动化



我们先来回顾一下上一篇提到的两个例子：

1. 如果第二天气温降低，发通知提醒我；
2. 每天晚上 21 点，在印象笔记中创建日记。

根据之前的知识我们知道，上边两个例子中， IFTTT 是 通过「轮询」的方式，在云端（也就是 IFTTT 的服务器上）调用相关服务的 API [2] 实现自动化的。那么这里就有一个问题，如果一个应用或服务没有 API 它可以通过 IFTTT 自动化吗?

我们通过一个例子看一下。

事例 1：每天下午六点将第二天的天气预报添加到 iOS 「日历」应用

按照惯例先分析一下这个需求：

- Trigger： 每天下午六点钟获取到第二天的天气信息。
- Action： 将天气信息写入到「日历」应用。

这个需求中， 我们可以通过 IFTTT 的 **Weather Underground** 获取第二天的天气信息。这一步应该比较简单，接下来就是将天气信息写入到「日历」应用中了。

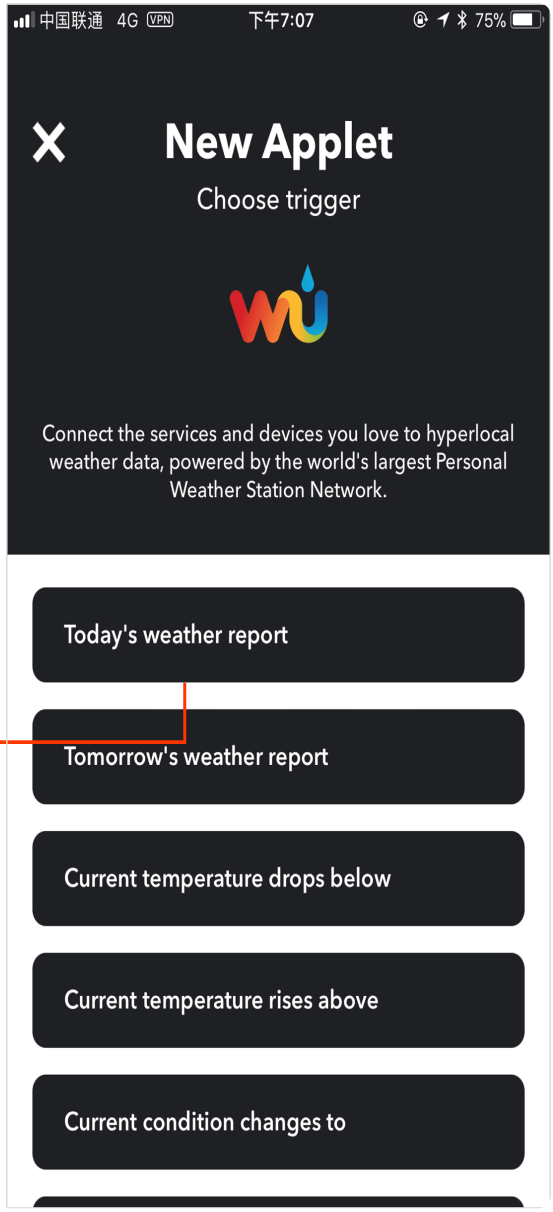
「日历」是 Apple 自家的应用，它没有提供 API，但是我们可以通过 IFTTT 的客户端获取到对「日历」应用的操作权限，从而将天气信息写入到「日历」应用。

设置 Trigger

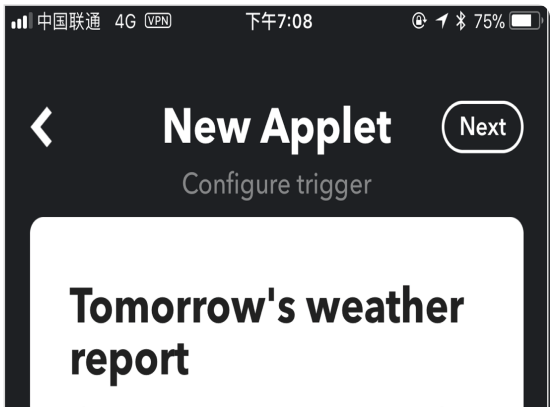
通过 **Weather Underground** 服务，在指定时间生成天气预报信息。

设置 Trigger：

1. 设置 Trigger：当生成第二天的天气信息后触发

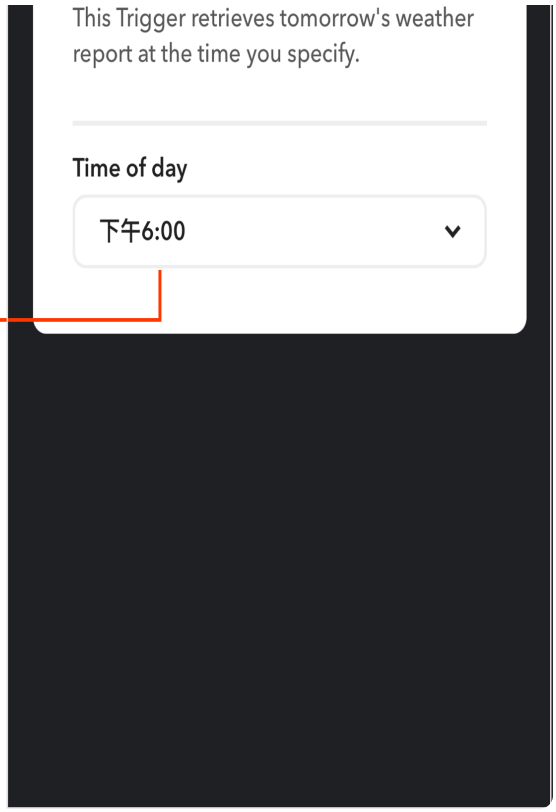


选择 Trigger 条件



设置 Trigger：

1. 设置天气预报生成时间

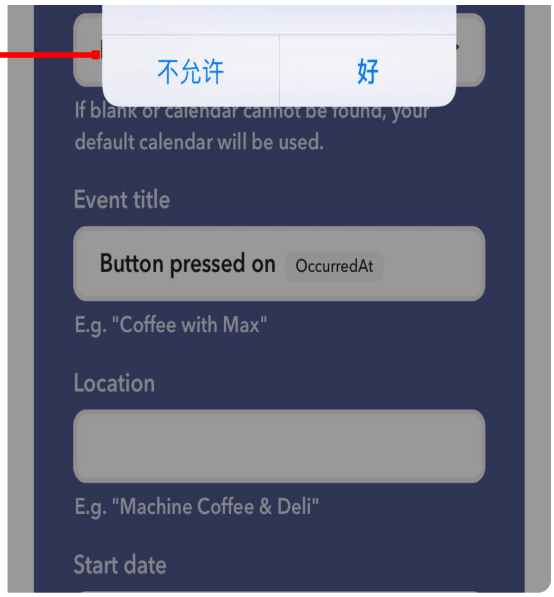


设置 Trigger 详情

设置 Action

创建第二天早上的日历事件，方便我们一早查看日程并了解天气信息，这里使用 **iOS Calendar** 服务。使用前先同意 IFTTT 获取到日历权限，如图：

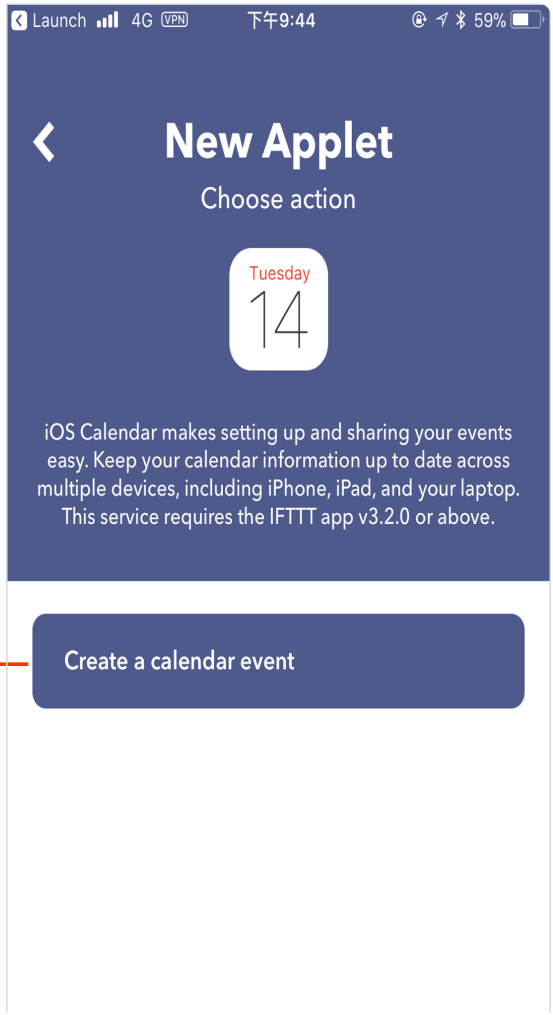
允许 IFTTT 应用访问日历



然后继续设置 Action，如图：

设置 Action：

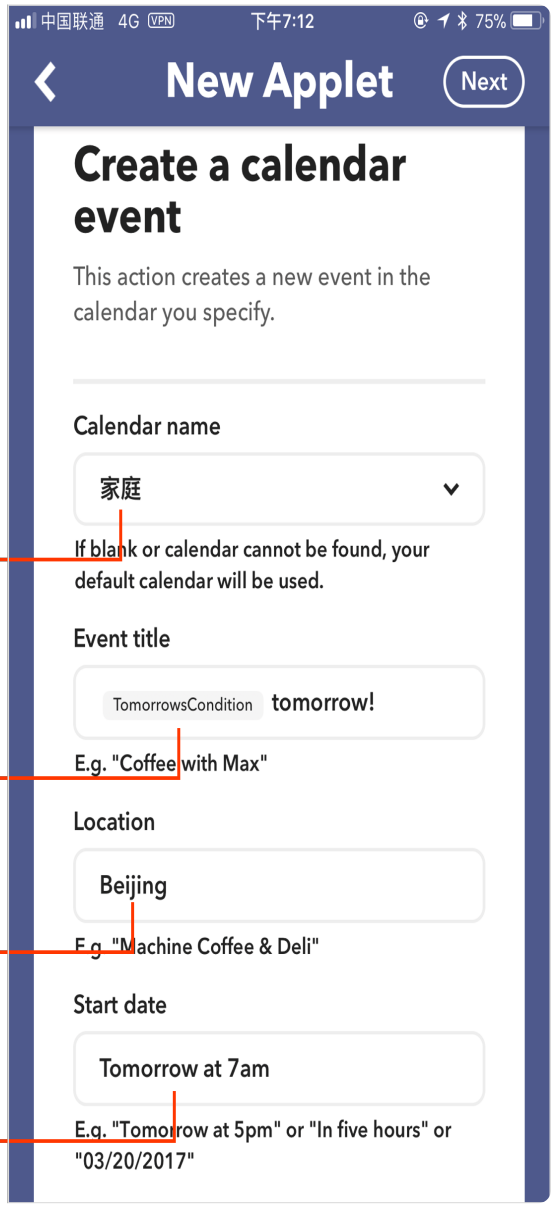
设置动作为「创建日历事件」



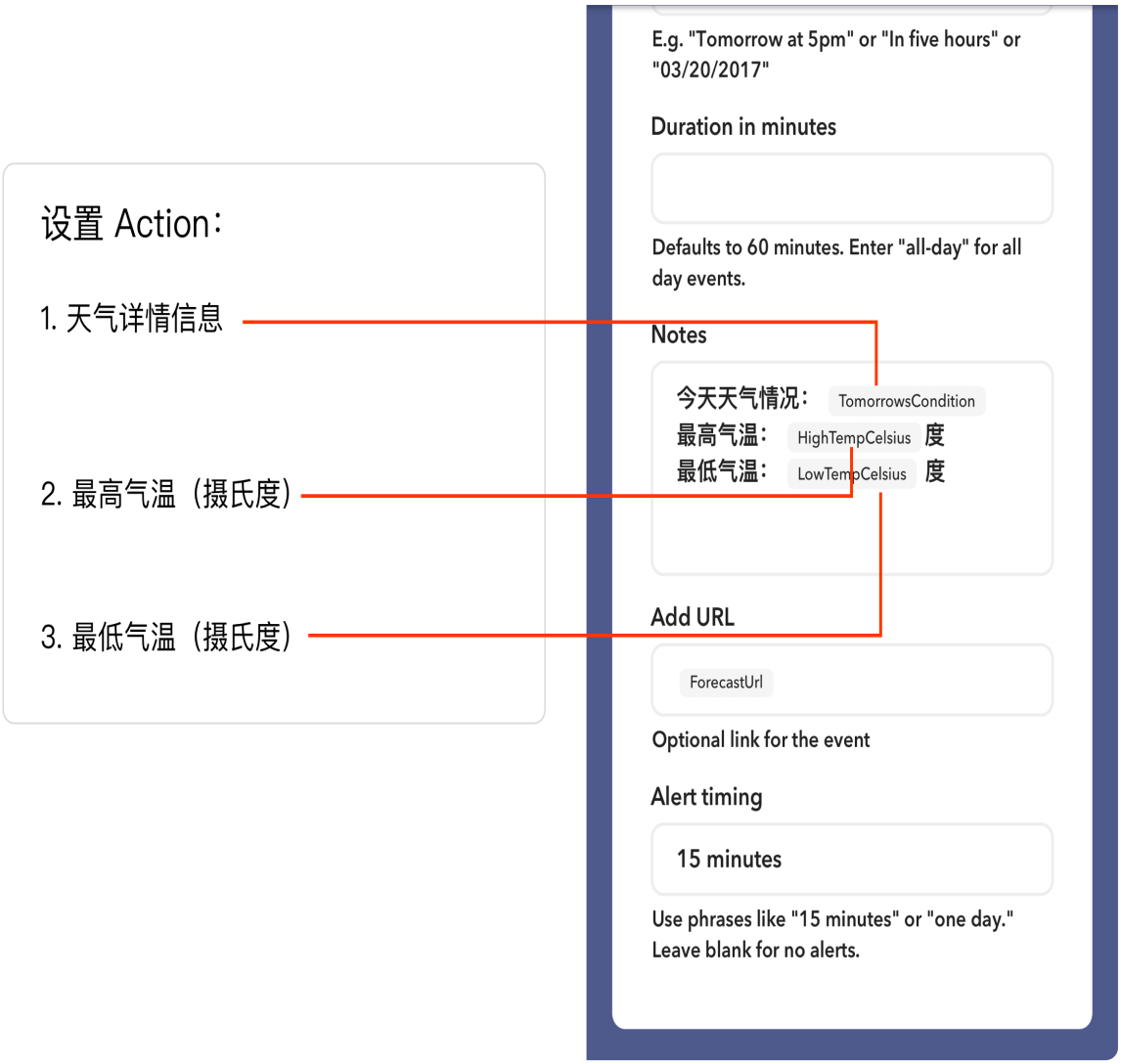
设置 Action 1

设置 Action：

1. 选择日历列表
2. 标题为第二天的天气信息
3. 填写你在 Weather 注册的地址
4. 日程开始时间为第二天早上7点



设置 Action 2



设置 Action 3

查看执行结果

在本例中，IFTTT 通过云端天气服务的 API，定时在下午 6 点获取到了第二天的天气信息，我们只要启动一次 IFTTT（或者将在后台的 IFTTT 运行），IFTTT 即可执行对应的动作将天气信息写入「日历」应用。

查看执行结果

我们知道在 IFTTT 中一个「自动化任务」是由 Trigger 和 Action 组成的，在事例 1 中，我们通过 IFTTT 客户端获得读取「日历」的权限，从而实现将天气信息写入「日历」应用中，这是一个典型的 Action，那么问题来了：

IFTTT 可以让没有 API 的本地化应用成为 Trigger 吗?

我们通过事例 2 看一下。

事例 2：自动将 Android 新短信 Pushover 推送到 iPhone

这个需求可能乍一看比较奇怪，按照惯例我们分析一下：

- Trigger：Android 手机收到新短信时被触发。

- Action：通过 Pushover 推送到 iPhone 手机。

看了这个需求，相信大家会有几个问题，我先解释一下：

问题 1：为什么是 Android 手机收到新短信推送到 iPhone，而不是反过来?

回答这个问题很简单，由于系统限制 iOS 系统即没有提供「短信」相关的 API，也没有提供读取「短信」的本地权限（换言之通过 IFTTT 客户端也无法读取短信）。所以 iPhone 手机收到短信这个条件无论如何是无法触发的。而 Android 系统由于其开放性，通过用户授权是可以读取「短信」信息的。

Tips：这里给 iPhone 双机用户提供一个「短信转发」的思路：手机设置里可以打开「信息」的「信息转发」功能，允许将收到的短信推送到其它已登录 iMessage 信息账户的设备，如图：

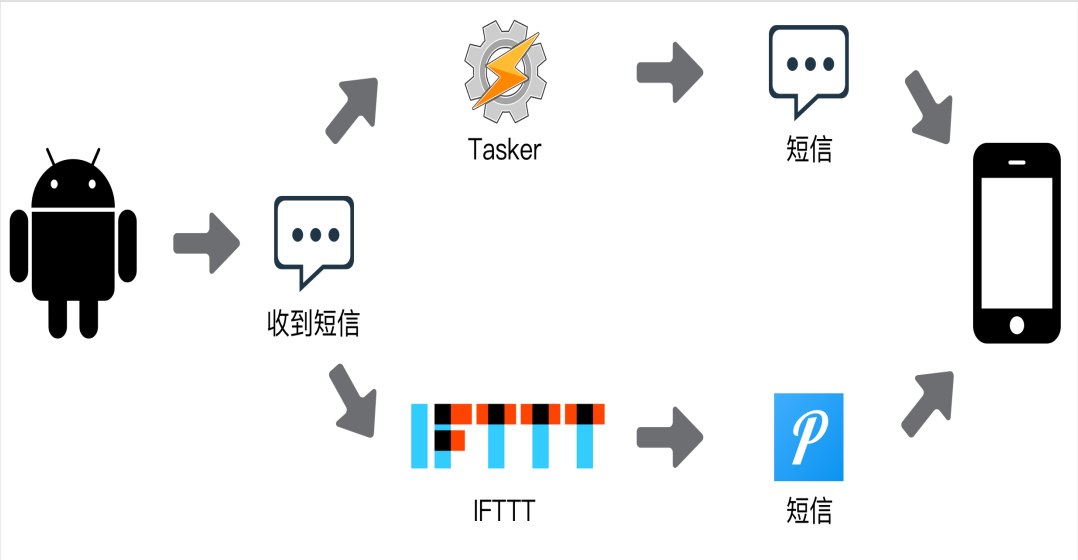
问题 2：为什么用 IFTTT 实现短信转发而不是别的应用比如 Tasker?

Tasker 是 Android 系统上类似 Workflow 的工具，通过设置条件和对应的动

作，实现自动化任务。这样看来，我们完全可以通过 Tasker 实现短信转发。

为什么要用 IFTTT 替代掉 Takser 呢？我有两个原因：

1. **免费**。通过 Tasker 转发短信的实现机制是，当 Tasker 发现有新的「短信」时，调用发短信的动作，将短信息转发给制定的手机号。所以它走的是「短信」通道，每转发一次都会产生费用。而 IFTTT 将短信通过 Pushover 服务推送到 iOS 上，只走网络流量，这个费用比短信低，如果手机在 Wi-Fi 环境下的话，成本几乎可以忽略不计。



2. **学习成本较低**。如果你是极客，一定会想到 Tasker 也可以通过调用 API 将短信发到我们的各种云服务上（比如 Telegram、Slack 等）。我认为这种方式比较麻烦，设置起来不容易，还需要多少懂一些编程知识，而 IFTTT 的优势就是它已经帮我们集成好了众多 API 服务，这样对用户是比较友好的，我们不需要关心细节（调试 API），只需要简单地动手设置就可以使用了。

问题 3：为什么要通过 Pushover 推送，而不是别的推送，比如 IFTTT 的通知？

既然是接收短信转发，那么它必须满足两个条件：

既然是接收短信转发，那么它必须满足两个条件：

1. 能够以「通知」的方式告诉我有新消息。
2. 点击「通知」可以查看消息。

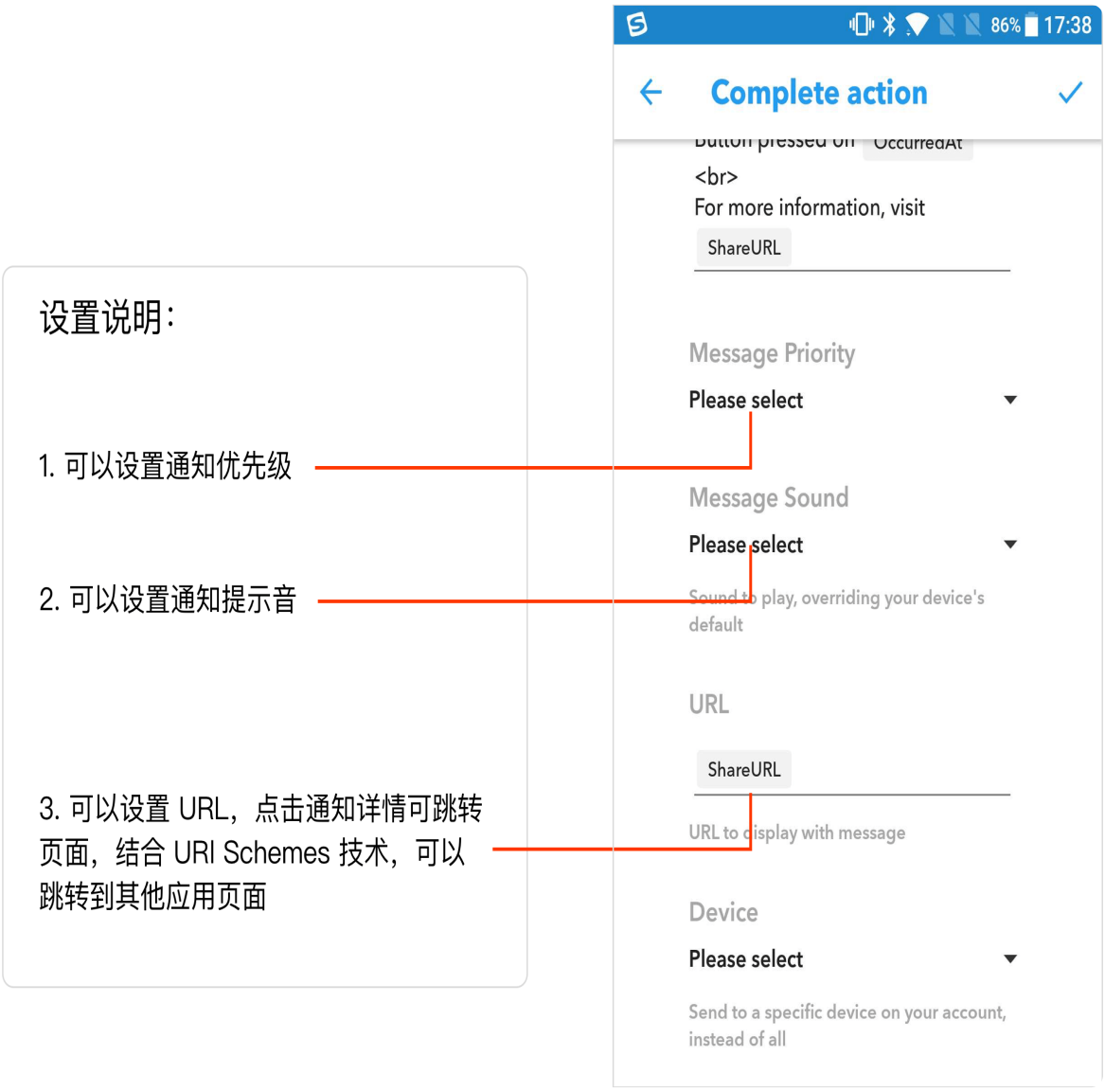
如果用 IFTTT 的 **Notifications** 获取作为短信转发的功能，第一条满足了，但是 IFTTT 的通知只能再通知界面下查看，且清除通知后，就再也无法查看了，这不符合第二条。而 Pushover 服务是可以在本地保存通知消息的，同时 Pushover 还是跨平台应用，也就是我们可以同时在 iPhone 和 iPad 上随时查看过往的短信息，而不用担心丢失。

所以这里选择 Pushover 实现短信转发。那么 Pushover 到底是什么服务呢?

Pushover： 高度自定义的推送通知服务

Pushover，是一个跨平台的推送通知服务，可以与 IFTTT，shell 脚本，网络监

控等服务一起使用，将推送通知发送到指定的设备。它提供了一套 API，只要调用 API 就可以给安装 Pushover 应用的手机推送通知，那么他的高度自定义都包括什么呢？看下图：



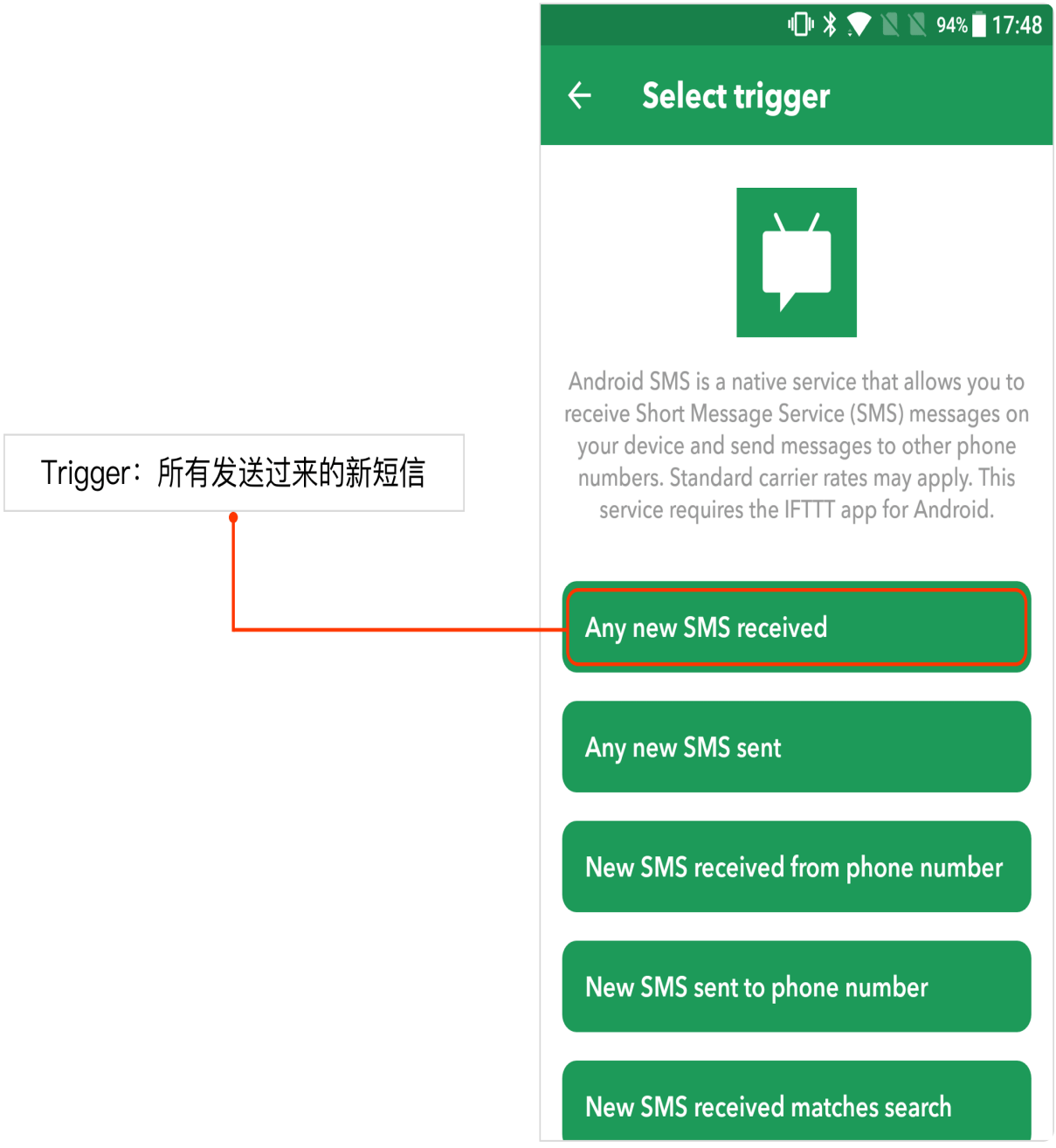
Pushover 设置

注：图中 URL 参数，可以利用 URL Schemes 技术跳转到手机上的其它应用中

接下来我们看一下他是如何实现的吧。

设置 Trigger（触发器）

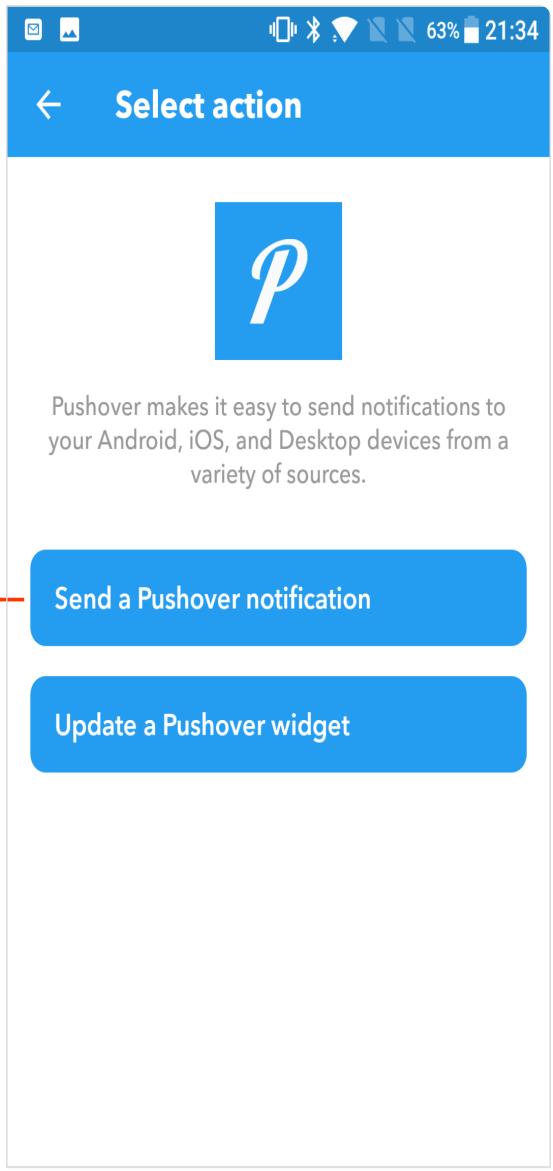
IFTTT 提供了 **Android SMS** 服务，支持读取 Android 手机上的短信，我们的触发条件是「当手机收到新的短信时」，Trigger 设置如图：



设置 Trigger

设置 Action（动作）

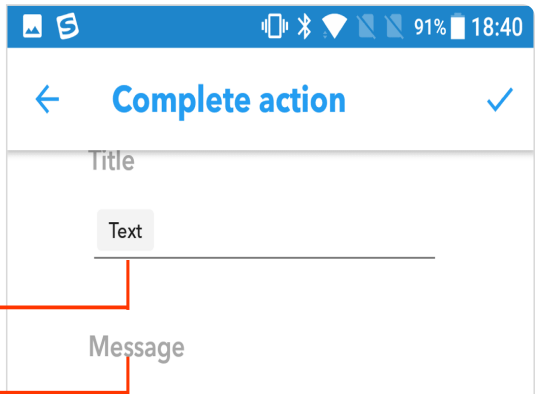
我们的动作是将收到的短信内容发送到 iPhone 的 Pushover 应用，Action 设置如图：



设置 Action：

设置动作为「发送通知到 Pushover 应用」

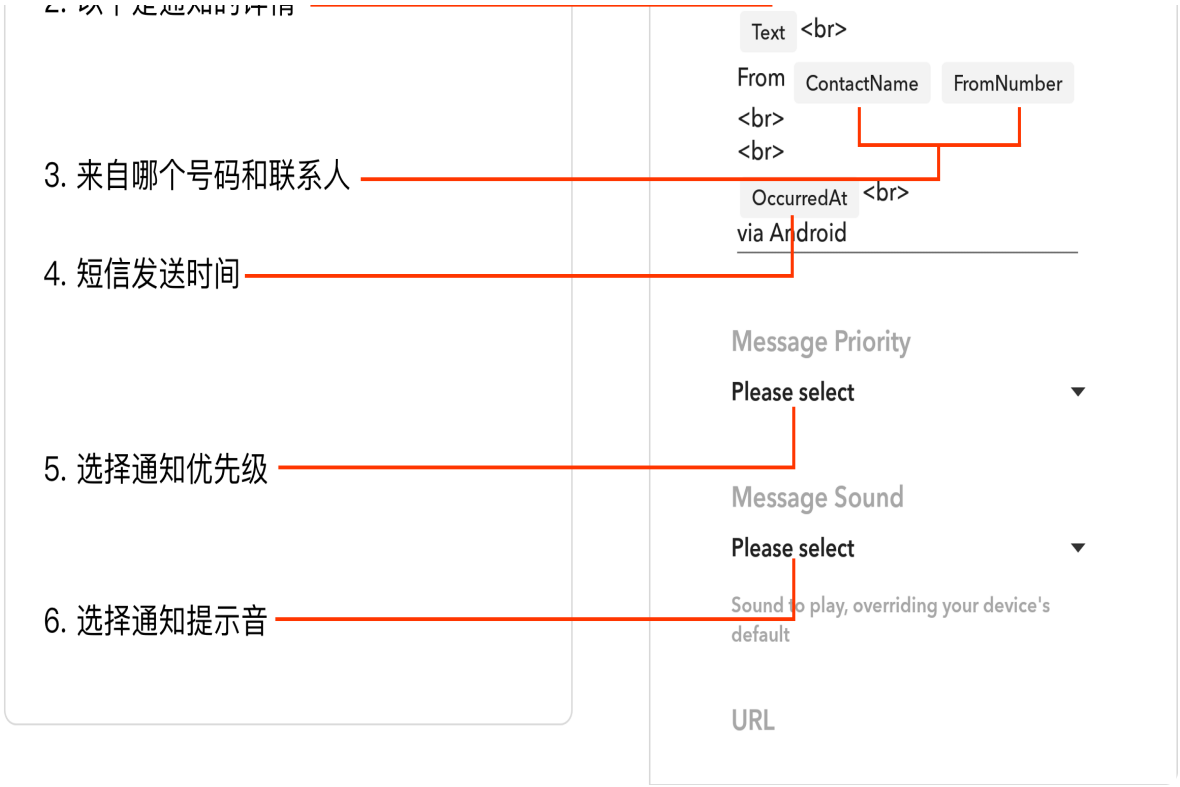
设置 Action 1



设置 Action 模板：

1. 通知的标题等于短信的详情

2. 以下是通知的详情



设置 Action 2

转发结果示意

转发结果示意

小结

在事例 2 中我们通过 IFTTT 获取到 Android 手机的短信权限，从而实现了当有新的短信息时自动转发到 iPhone，通过本例我们证实了刚才提出的问题：**IFTTT 可以让没有 API 的本地化应用成为 Trigger**！

接下来我们再看一下 IFTTT 客户端的第二个作用。

02. 通过 Widget，手动触发 IFTTT 使其完成自动化任务

Widget，iOS 通知中心小插件，在插件上我们可以显示列表进行快速查看，典型的场景是日历插件，可以快速查看当日日程。同时我们还可以在插件上有按钮，有了按钮我们可以通过点击按钮进行一些快速操作，比如「跳转到某个应用」，再比如「快速记录内容」等。

一直以来我们都是用 IFTTT 实现「如果 A 完成了事情 1，那么就让 B 完成事情 2」这样的任务，了解 IFTTT 的本质就会发现，IFTTT 集成了大量服务的 API，我们只需要简单的配置就可以使用这些服务。

看到这里想必你也会想到，如果 IFTTT 能够和 Widget 结合起来，我们应该可以实现许多有意思的功能，接下来我们通过几个事例看一下如何实现。

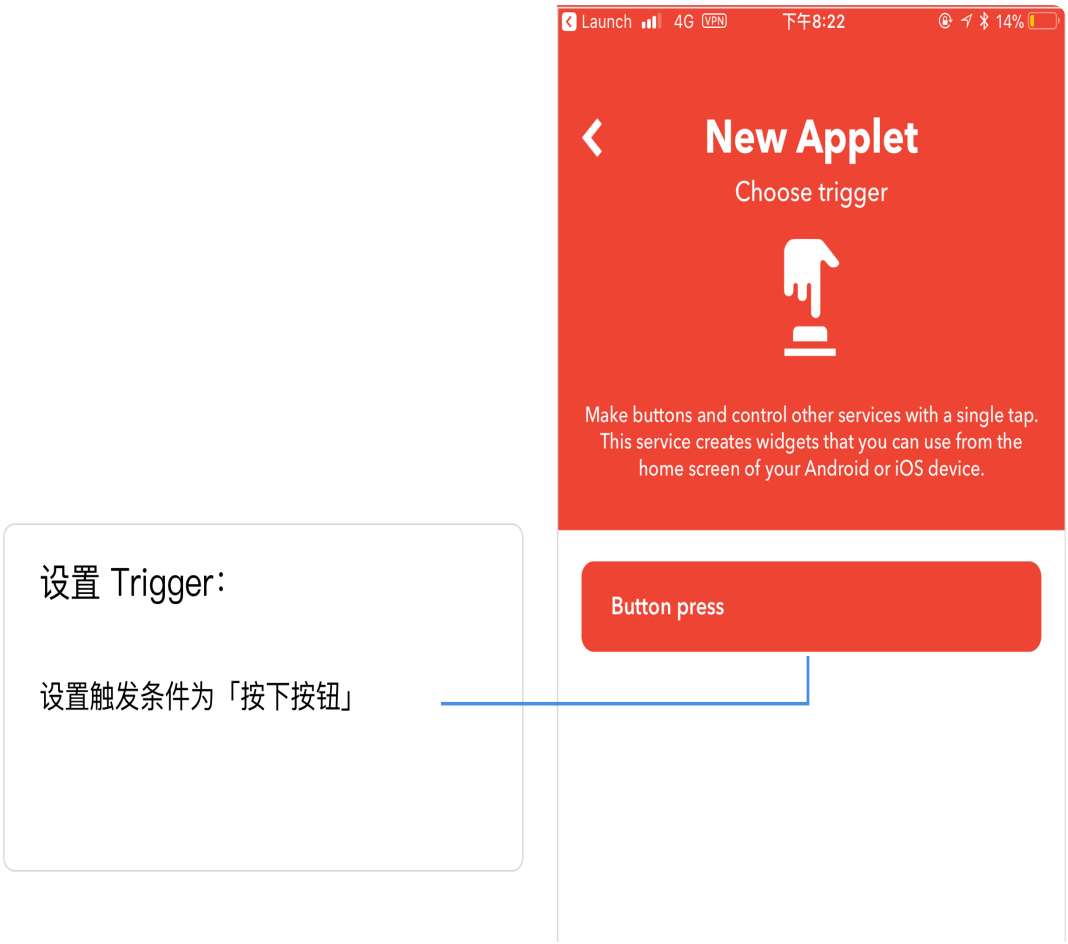
事例 3：通过 IFTTT 记录理发时间到 Google Sheets

我们每次理发的时候都会被问到「上次是什么时候理的发？」此类问题，相信大多数人都记不住的。我们不妨每次理发完成后记录一下时间（记录到 Google Sheets[3]），这样下次再被问到的时候可以很详细的回答。（by @JailbreakHum）

这个动作看起来比较小，可是仔细分析步骤也比较繁琐，我们每次要打开 Google Sheet，新建一条记录，记录当前时间，整个流程下来要好几步。

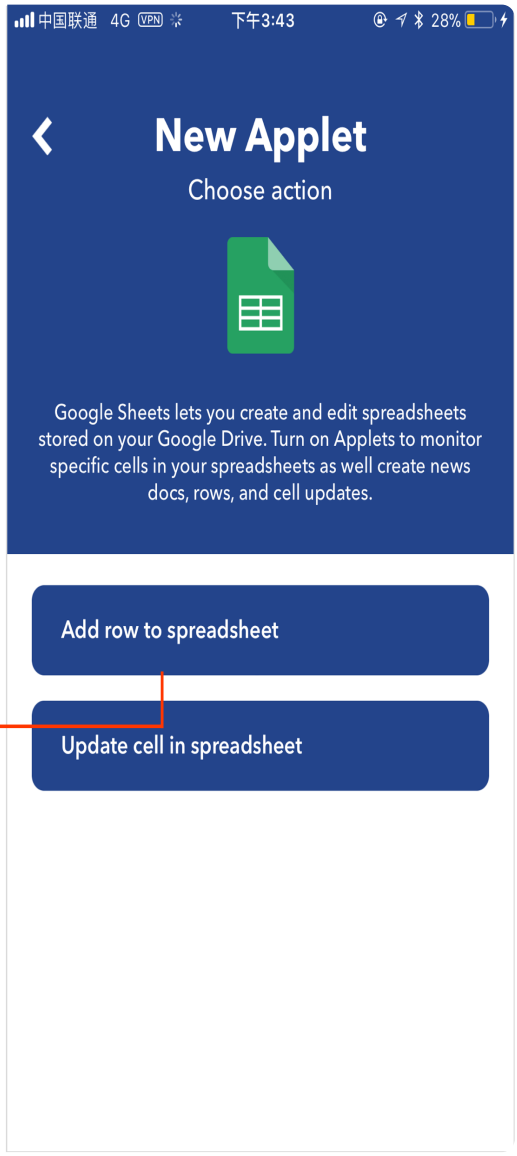
IFTTT 里有一类动作叫 **Button**，这个 Button 实际上是借助 iOS 的 Widget 实现的，简单说我们通过 IFTTT 的 Button 服务，一键就能完成一些操作，比如本例中记录理发时间。

1. 创建 Trigger：添加 IFTTT **Button widget** 服务，触发条件为「按下 Widget 按钮」



设置 Trigger

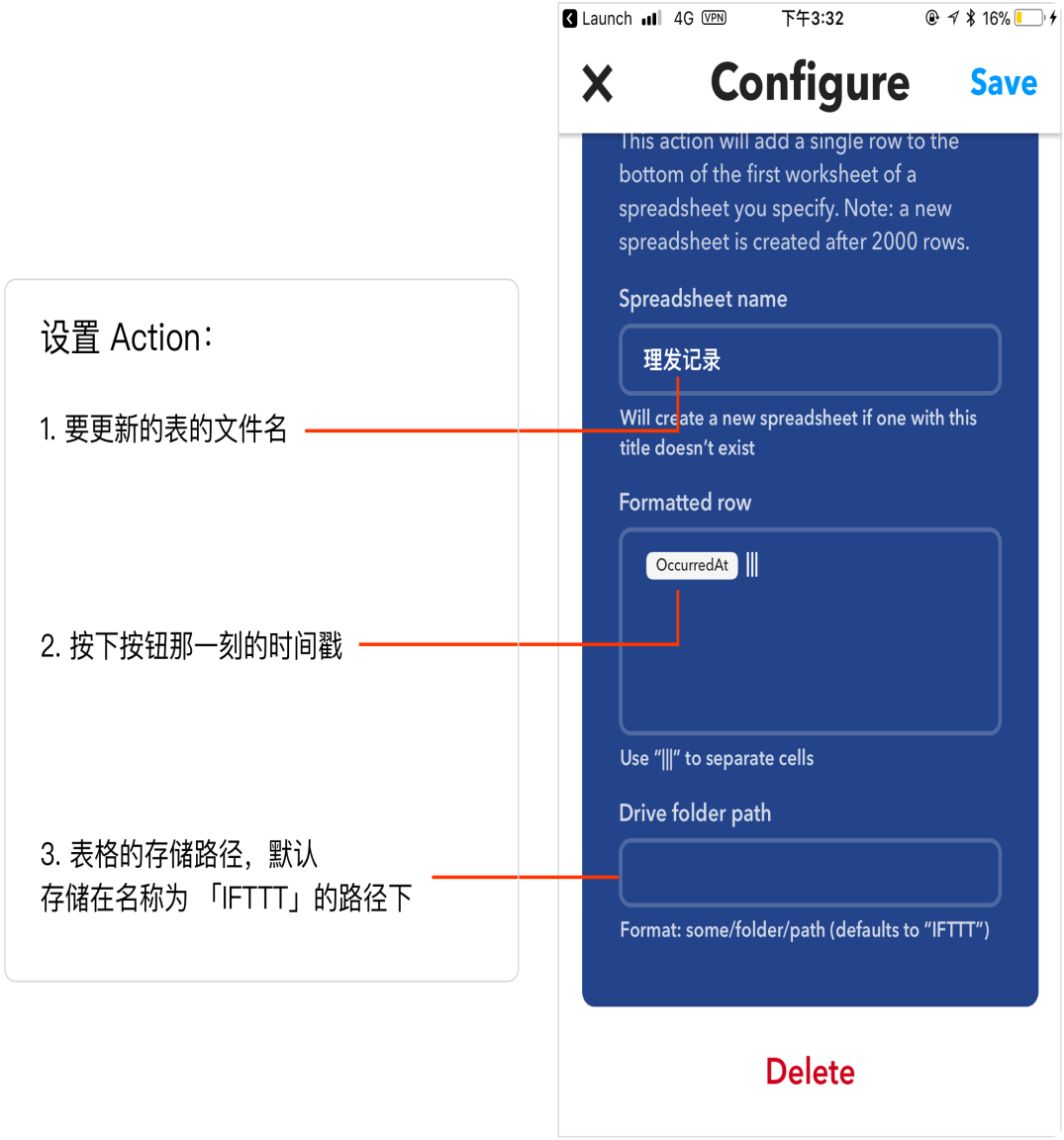
2. 设置 Action： 添加 IFTTT **Google Sheets** 服务，动作为「将一条记录添加到 Google Sheets 中」


中国联通 4G VPN 下午3:43 28%
New Applet
Choose action
Google Sheets lets you create and edit spreadsheets stored on your Google Drive. Turn on Applets to monitor specific cells in your spreadsheets as well create news docs, rows, and cell updates.
Add row to spreadsheet
Update cell in spreadsheet

设置 Action:

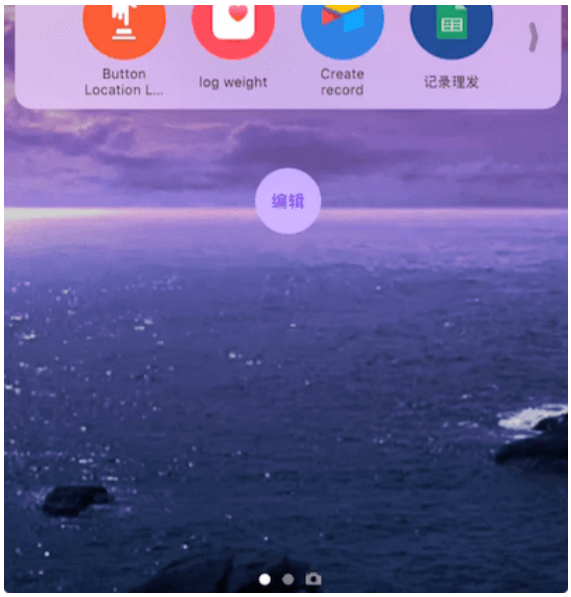
设置动作为「添加一条新纪录」

设置 Action 1



设置 Action 2

3. 执行动作，记录理发时间：

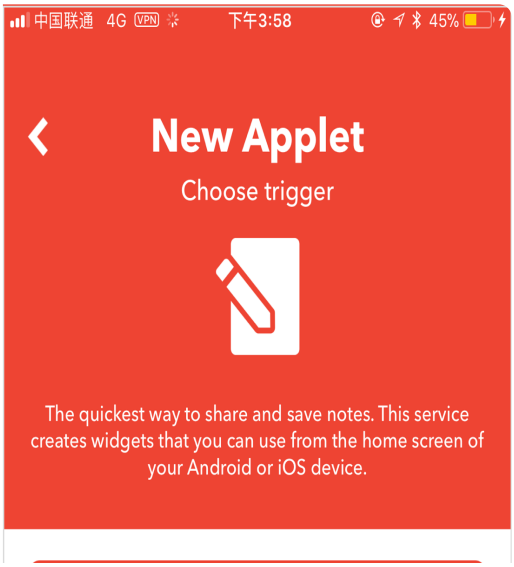


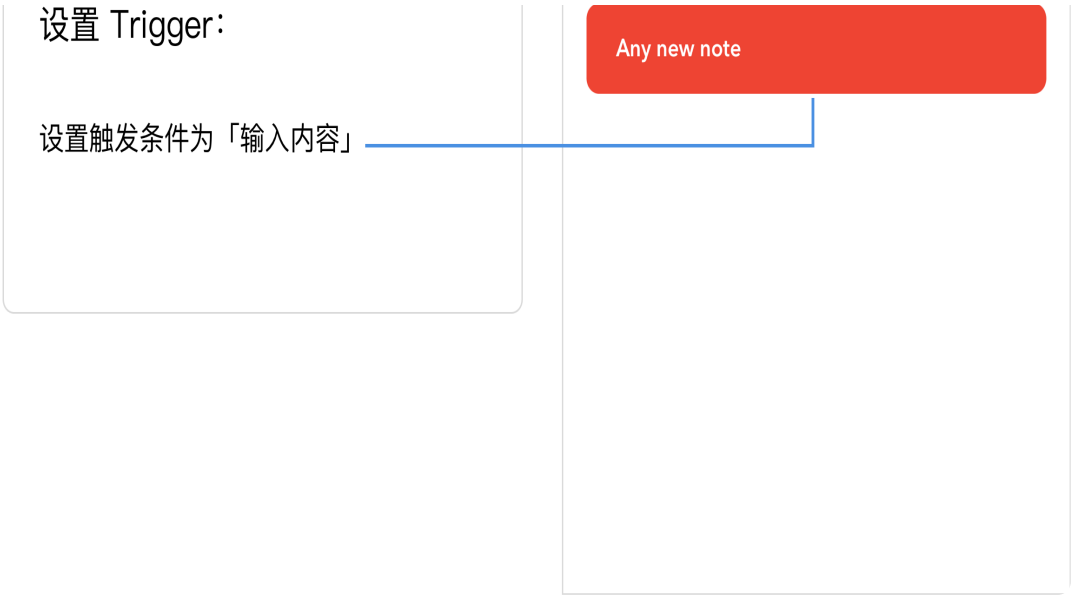
事例 4：快速记录灵感保存到 Evernote

iOS 上关于灵感记录类软件有不少，如果你是 Evernote 和 IFTTT 用户的话，可以使用 IFTTT 通知中心小插件实现快速记录灵感的功能。

这里要说明一下，Evernote 应用自己也提供了 Widget 插件快速记录内容，之所以不使用不是因为 Evernote 插件不好用，而是因为我习惯了在 IFTTT 中统一管理和使用插件。

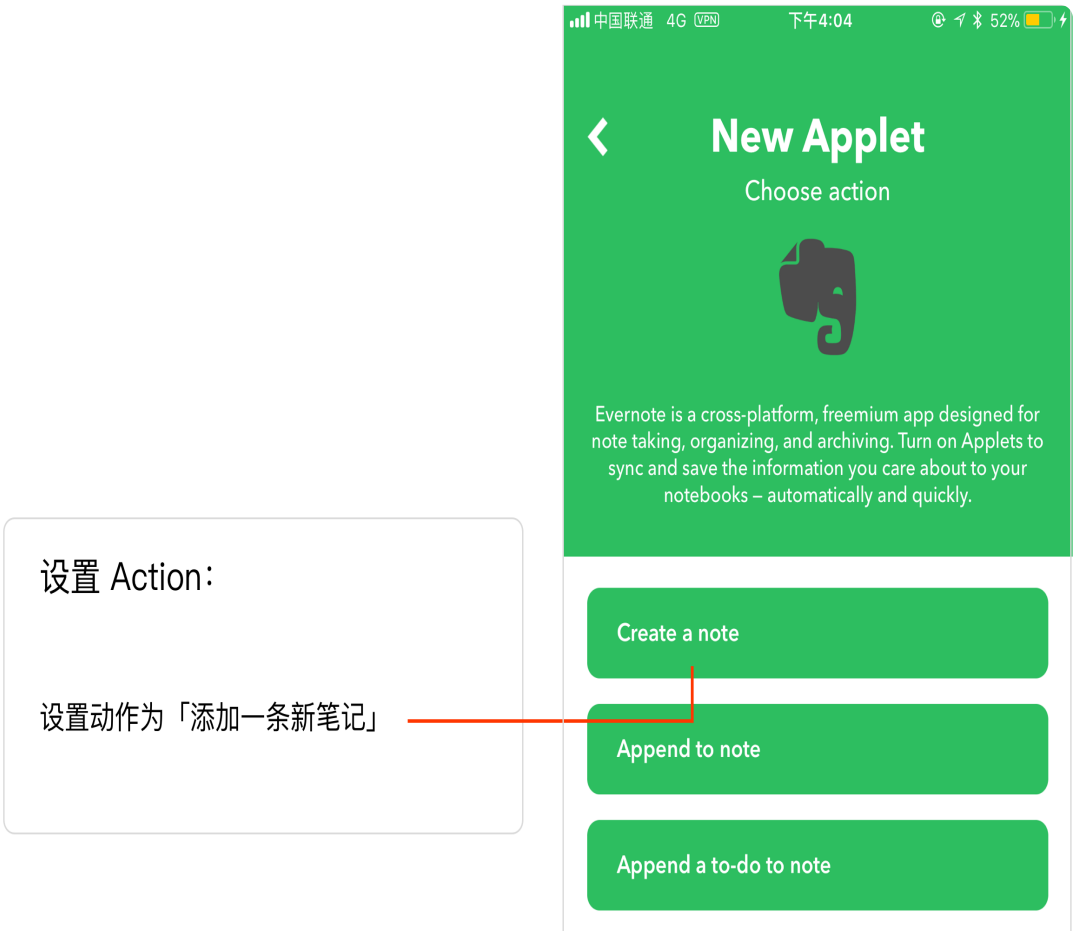
1. 创建 Trigger：添加 IFTTT **Note widget** 服务，触发条件为「输入内容」

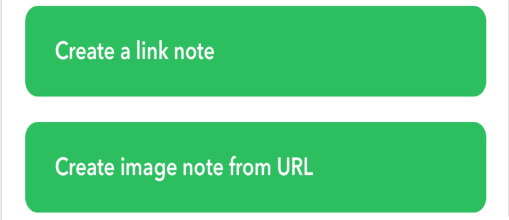




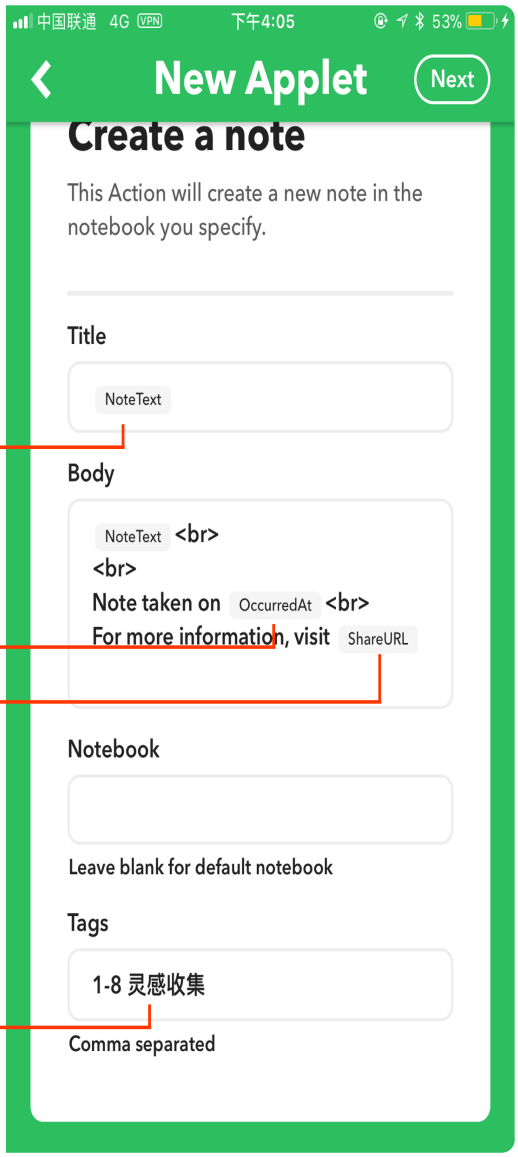
设置 Trigger

2. 设置 Action：添加 IFTTT **Evernote** 服务，内容为「保存文本到 Evernote」


设置 Action：
设置动作为「添加一条新笔记」
中国联通 4G VPN 下午4:04 52%
New Applet
Choose action
Evernote is a cross-platform, freemium app designed for note taking, organizing, and archiving. Turn on Applets to sync and save the information you care about to your notebooks – automatically and quickly.
Create a note
Append to note
Append a to-do to note



设置 Action 1



设置 Action：

1. 输入的灵感内容
2. 灵感记录的创建时间
3. 灵感记录的分享地址
4. 设置笔记的标签

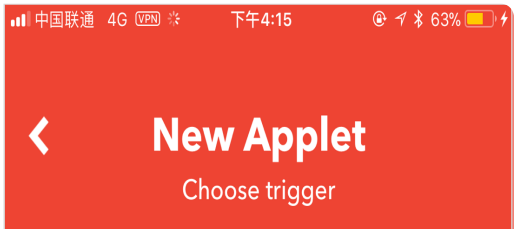
设置 Action 2

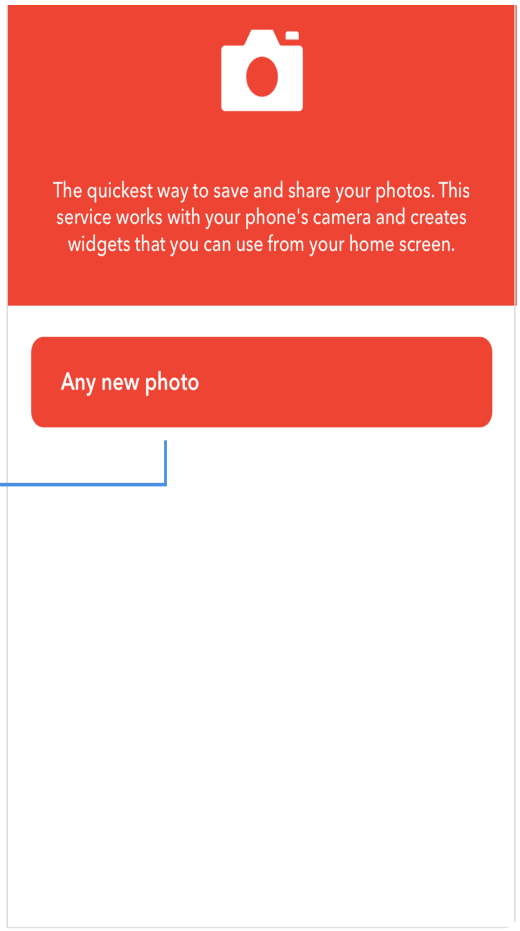
3. 执行动作，记录灵感：

事例 5：快速拍照保存到 Dropbox 指定目录

说到灵感记录，有一个减少输入的方法就是把当前产生灵感的场景拍照记录下来，等到有时间了再翻看照片来整理灵感，关于这个使用体验貌似也有 App 在做，这里王掌柜提供一个通过 IFTTT Widget 一键拍照自动存储的方式来实现快速记录灵感。

1. 创建 Trigger：添加 IFTTT **Camera widget** 服务，触发条件为「按下 Widget 按钮」



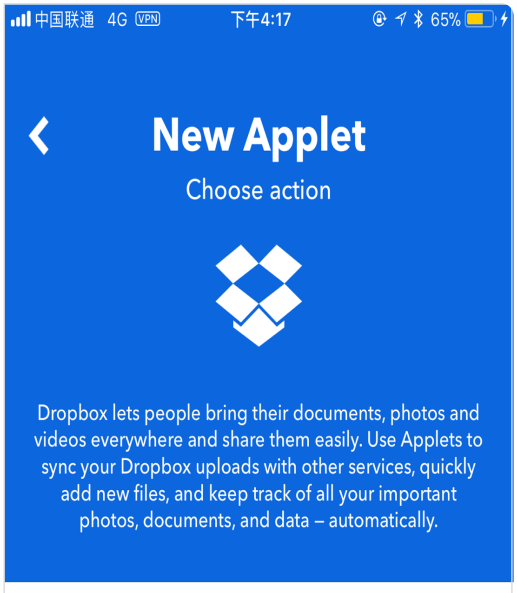

The quickest way to save and share your photos. This service works with your phone's camera and creates widgets that you can use from your home screen.
Any new photo

设置 Trigger：

设置触发条件为「按下按钮拍照」

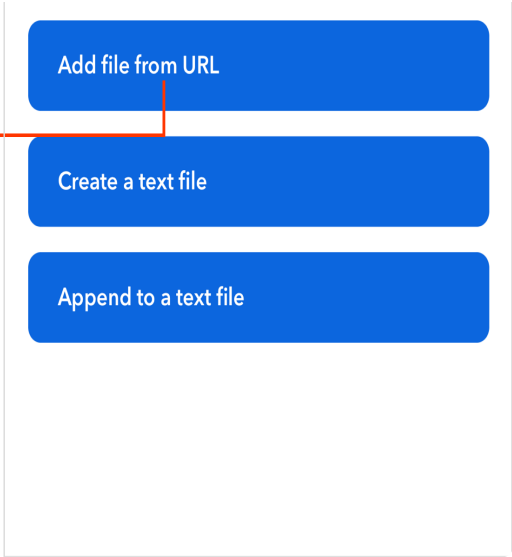
设置 Trigger

2. 设置 Action： 添加 IFTTT **Dropbox** 服务，内容为「拍照并保存到 Dropbox 指定目录」


中国联通 4G VPN
下午4:17
65%
New Applet
Choose action
Dropbox lets people bring their documents, photos and videos everywhere and share them easily. Use Applets to sync your Dropbox uploads with other services, quickly add new files, and keep track of all your important photos, documents, and data – automatically.

设置 Action：

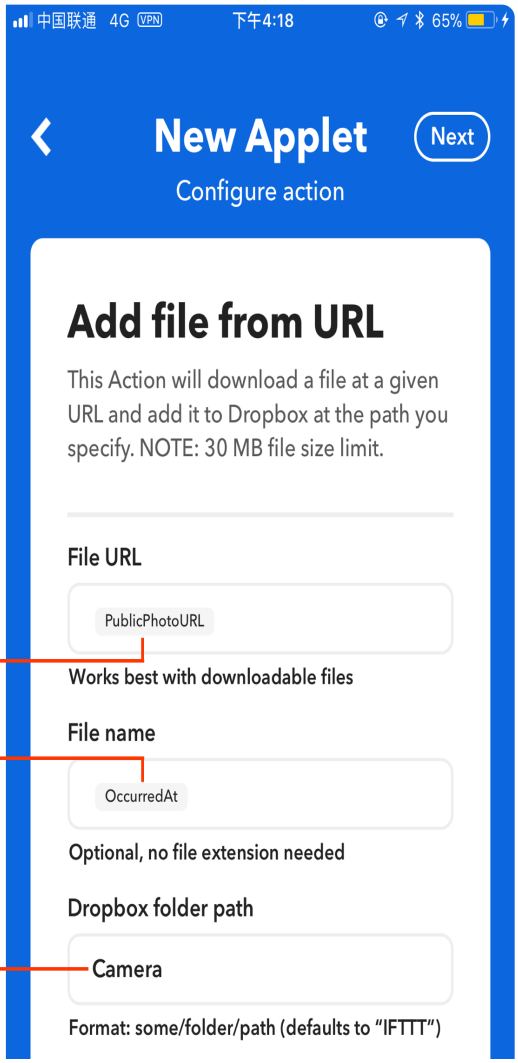
设置动作为「根据 URL 上传文件」



设置 Action 1

设置 Action：

1. 上一步拍照的照片 URL
2. 拍摄灵感照片的时间
3. 照片在 Dropbox 的存储路径



设置 Action 2

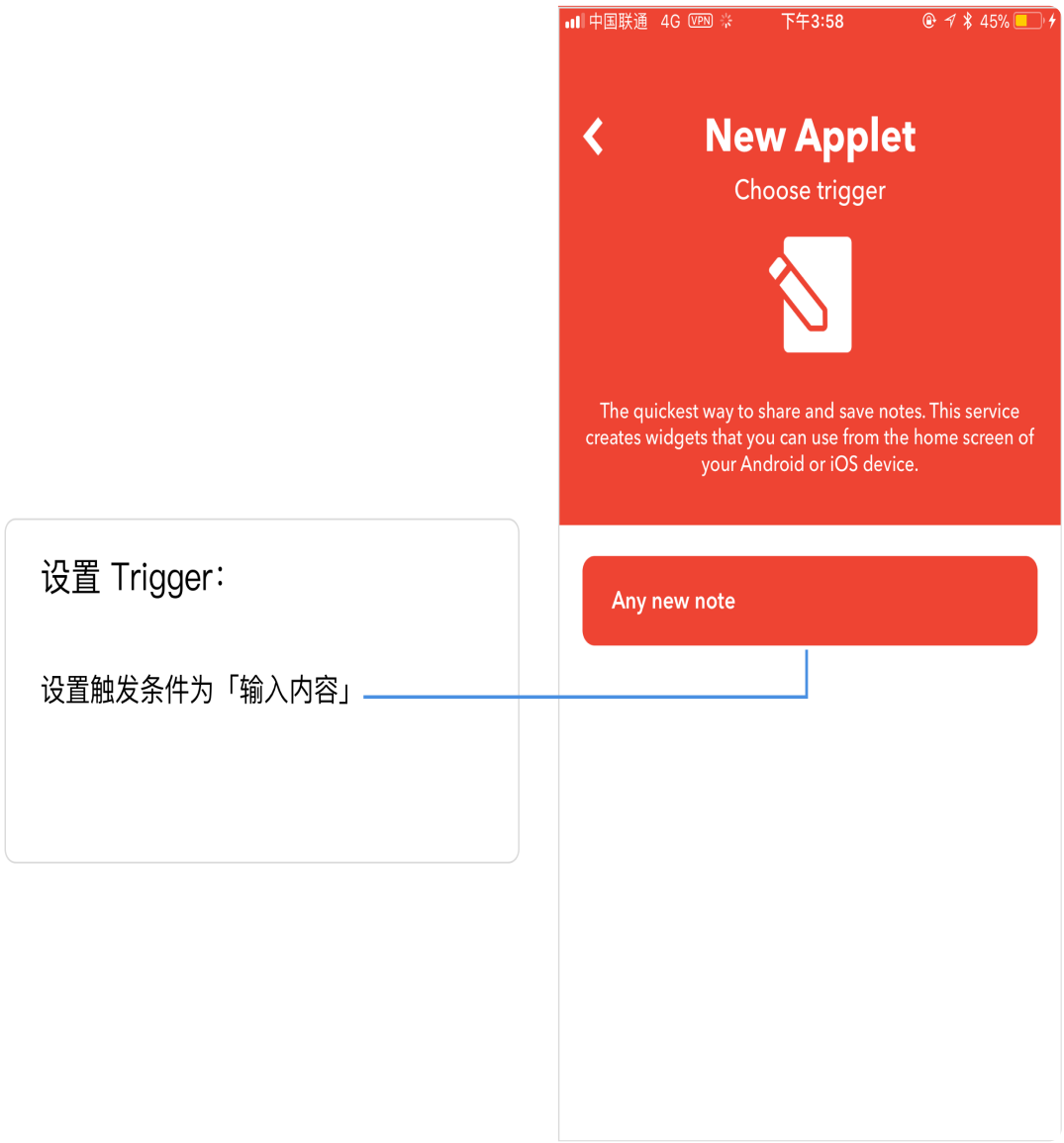
3. 执行动作，拍照并自动保存灵感图片

事例 6：快速录入体重到 iOS 健康应用里

iOS 「健康」应用提供了我们对健康数据的管理，许多服务和应用都连接到「健康」应用，写入或者读取健康信息。在我们日常中**体重信息**是一个许多人都关心的，也是非常容易测量的数据，现在我们可以通过 IFTTT 的 Widget 快速录入体重到「健康」应用。从而提高我们记录体重的效率。

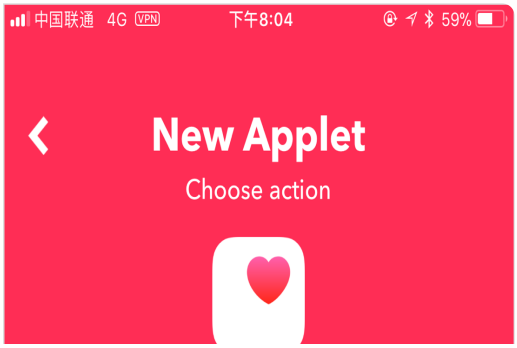
1. 设置 Trigger：添加 IFTTT **Note widget** 服务，触发条件为「按下 Widget 按钮」

按钮」


设置 Trigger：
设置触发条件为「输入内容」
中国联通 4G VPN 下午3:58 45%
New Applet
Choose trigger
The quickest way to share and save notes. This service creates widgets that you can use from the home screen of your Android or iOS device.
Any new note

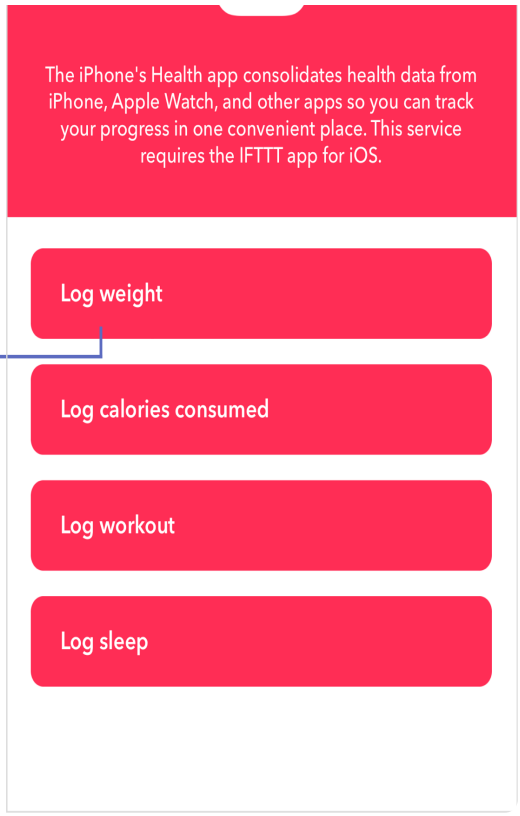
设置 Trigger

2. 设置 Action：添加 IFTTT **iOS Health** 服务，内容为「保存体重信息到健康应用」。



设置 Action：

设置动作为「记录体重到健康应用」

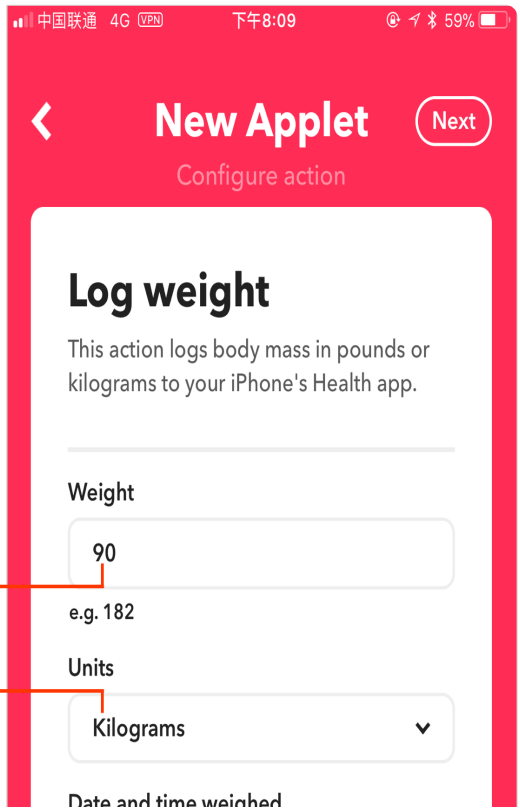


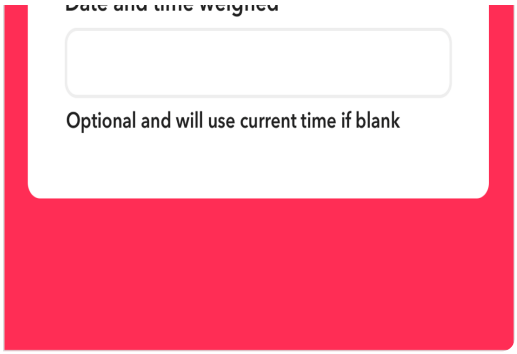
设置 Action 1

设置 Action：

1. 输入体重示例（随便输入数字即可）

2. 单位选择公斤





设置 Action 2

3. 执行动作，保存体重信息

03. 综合应用：IFTTT 实现我的照片存储方案

刚才的事例中我们已经了解了 IFTTT 客户端的作用，它可以让没有 API 的本地

刚才的事例中我们已经了解了 IFTTT 客户端的作用，它可以让没有 API 的本地应用实现自动化，接下来我们看一个稍微复杂的例子，在例子中 IFTTT 结合了 Workflow、Hazel 等应用，同时还能很好的提现 IFTTT 客户端的作用，这个事例也是我自己跑了很久的一个自动化任务。

很多人都有手机拍照的习惯，每天我们都会用手机拍摄很多照片，大家也会把照片分享到微博、微信、Instgaram 等平台，我自己也是一个手机拍照爱好者，而且是一个典型的 Android 和 iOS 双机党，我最大的困扰就是照片的日常管理，「如何把分散到两个平台的照片集中统一管理以及备份？」

首先总结一下我的需求：

1. 我希望随时都随地打开 iPhone 能够看到我在 iPhone 和 Android 手机上拍的所有照片；
2. 我希望我所有的照片在 iCloud 和 Dropbox 都有备份。

为什么会有这个需求呢?

第一，由于 iCloud 和 iPhone 照片其实是同步机制，也就是你在手机上删除一张照片，iCloud 就没了，所以严格意义说 iCloud 不算一个备份，所以我才考虑将照片另存一份到 Dropbox 中，完整的保留我所有的照片不留遗憾。而且，Dropbox 的稳定性也是业界有口皆碑的，存在那里会比较放心。

第二，根据存储 321 原则[4]，照片至少有两个备份，所以最终确定 iCloud 和 Dropbox 各自备份一次。

我的实现方案：

1. iOS 设备拍照，通过 IFTTT 存储到 Dropbox，做到 iCloud（iOS 默认存储） 和 Dropbox 双重备份。
2. Android 设备拍照，通过 IFTTT 存储到 Dropbox 做备份。

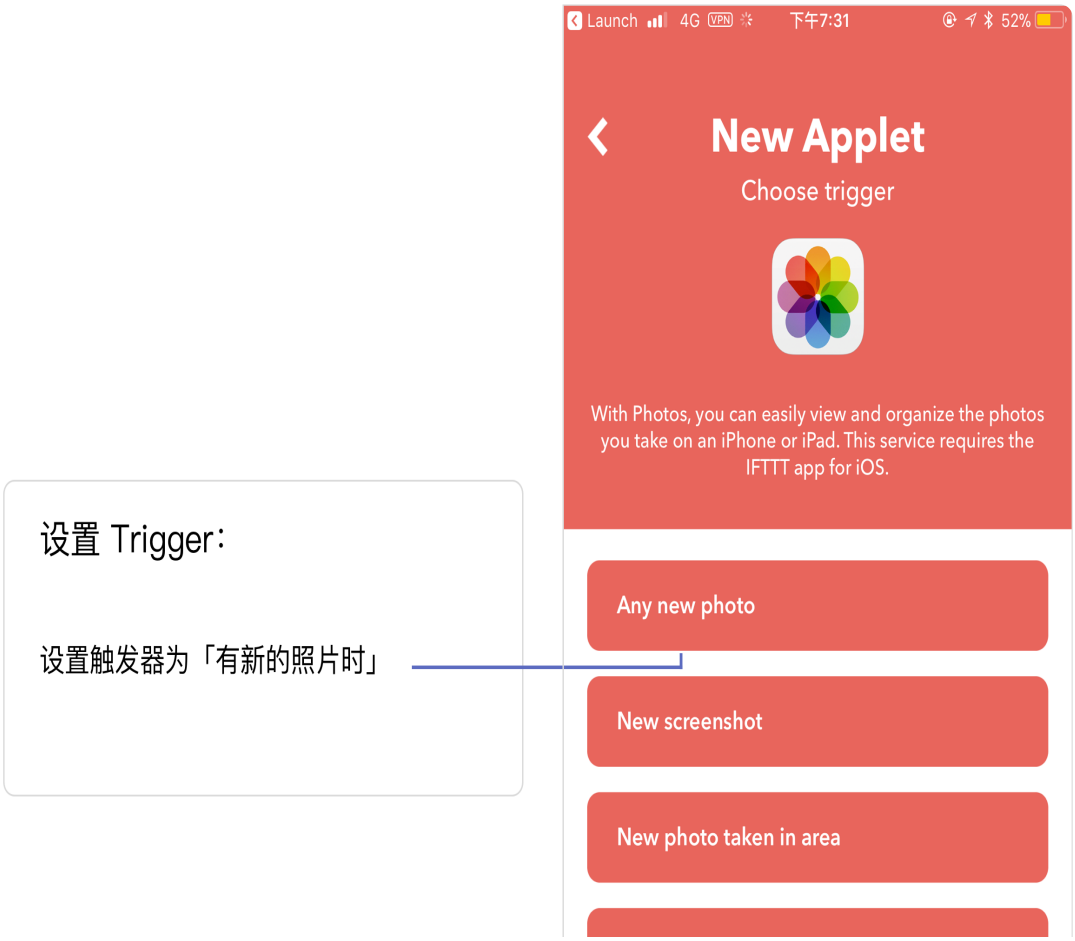
3. 通过 macOS 的 Dropbox 和 Hazel，将 Android 存储在 Dropbox 的照片导入到 macOS 的「照片」应用，利用 iCloud 同步，很快我的 iPhone 中就可以看到 Android 手机拍摄的照片了。
4. 最终所有照片在 iCloud 和 Dropbox 各有一份存储。并且可以在 iPhone 手机上查看所有照片。

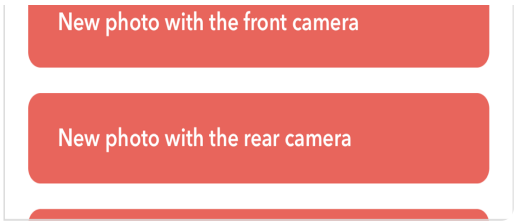
这是一个迄今为止我们讲到的最复杂的一个例子，大家也不用担心看不懂，我尽量用简单的话语描述，大家可以跟着我一起实现这个流程。

第一部分：IFTTT 监控 iPhone 添加新照片，保存到 Dropbox

创建自动化任务，实现 iOS 照片自动备份到 Dropbox

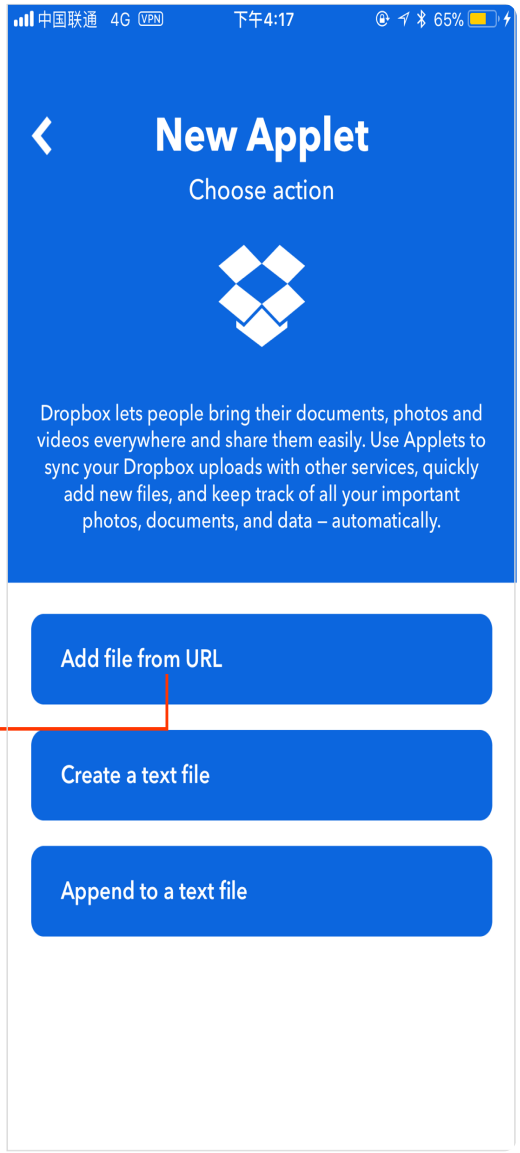
1. 创建 Trigger：当 IFTTT 发现有一张新照片的时候。





设置 Trigger

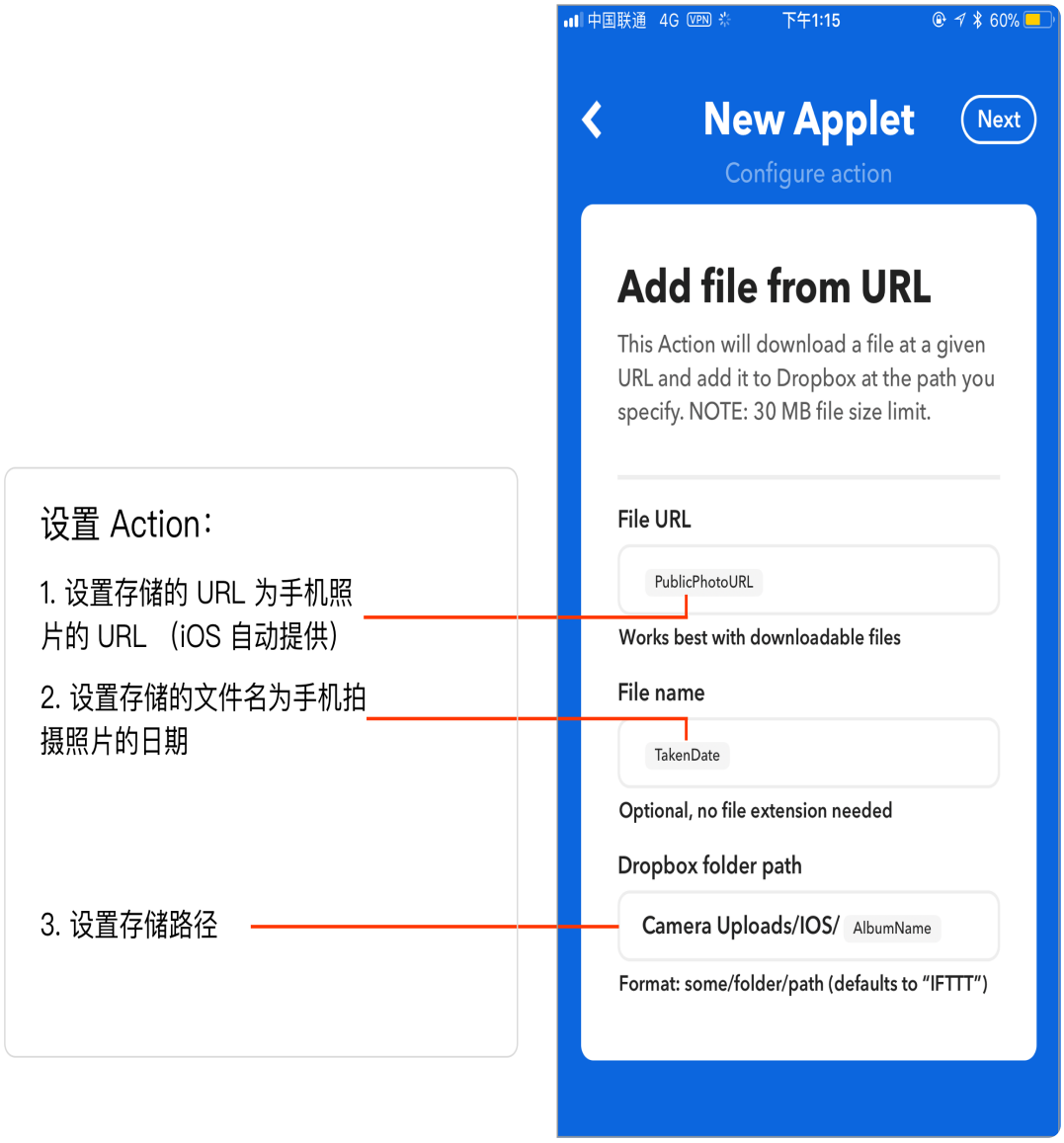
2. 创建 Action：将新照片同步到 Dropbox 的 `Dropbox/Camera Uploads/IOS` 中。


中国联通 4G VPN 下午4:17 65%
New Applet
Choose action
Dropbox lets people bring their documents, photos and videos everywhere and share them easily. Use Applets to sync your Dropbox uploads with other services, quickly add new files, and keep track of all your important photos, documents, and data – automatically.
Add file from URL
Create a text file
Append to a text file

设置 Action：

设置动作为「根据 URL 上传文件」

设置 Action 1



设置 Action 2

iOS 拍一张照片，打开 IFTTT App 使得条件被动触发

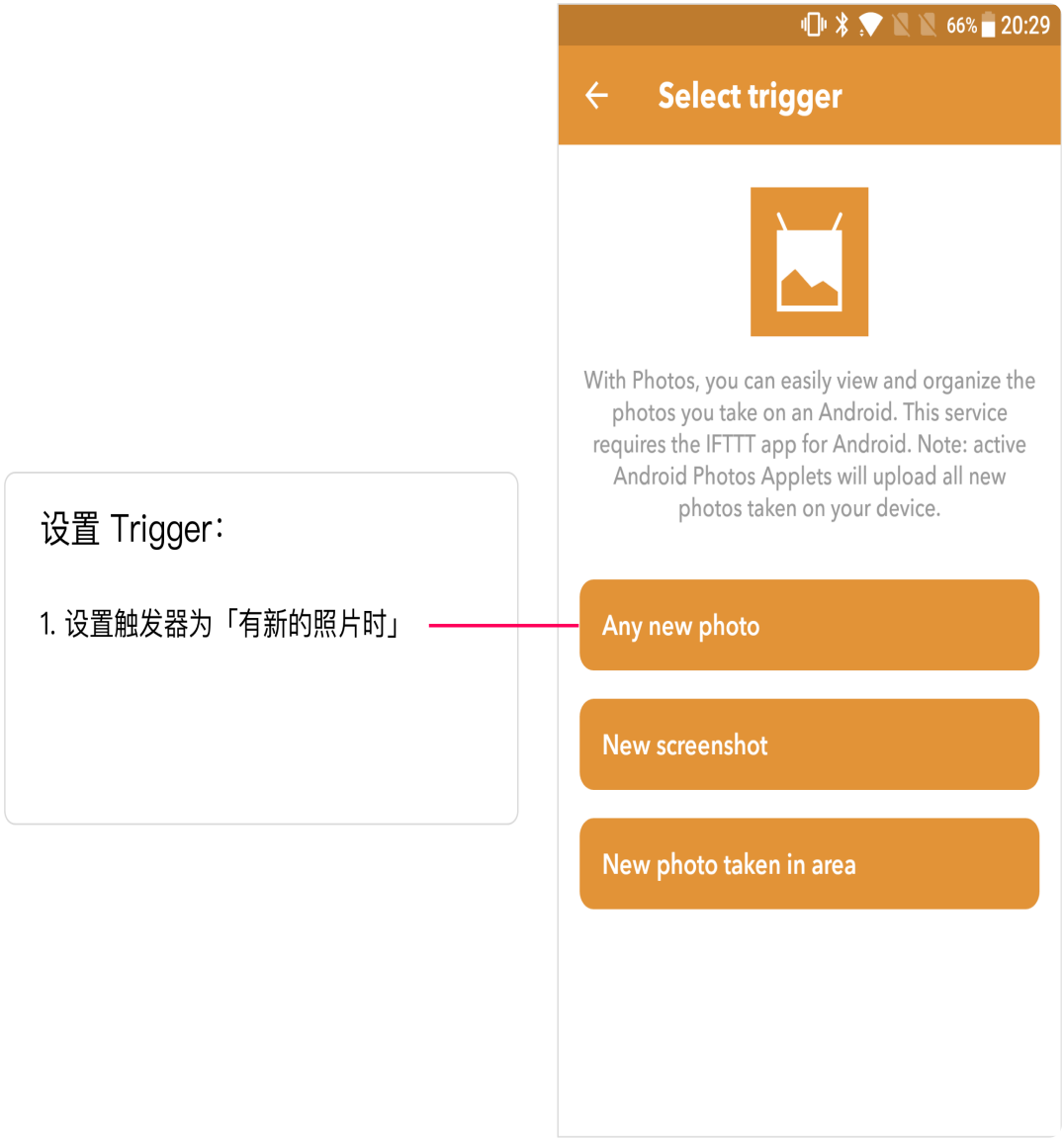
1. 打开iOS 相机 App 拍一张照片。
2. 打开 IFTTT App，会自动发现新加入的照片，然后启动上传动作。

第二部分：IFTTT 监控 Android 手机添加新照片，保存到

Dropbox

创建自动化任务，实现 Android 手机拍照自动备份到 Dropbox

1. 创建 Trigger：当 IFTTT 发现有一张新照片的时候。

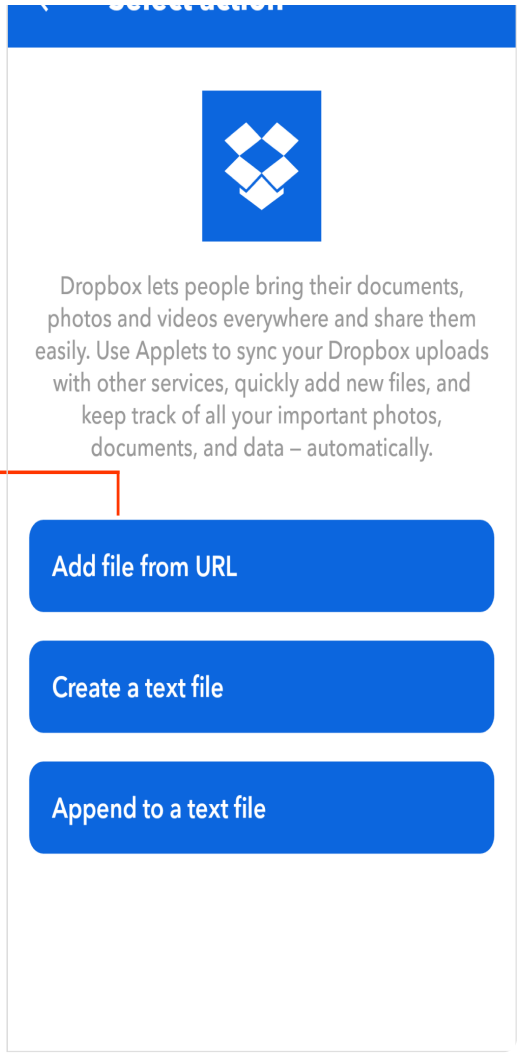

设置 Trigger：
1. 设置触发器为「有新的照片时」
66% 20:29
Select trigger
With Photos, you can easily view and organize the photos you take on an Android. This service requires the IFTTT app for Android. Note: active Android Photos Applets will upload all new photos taken on your device.
Any new photo
New screenshot
New photo taken in area

设置 Trigger

2. 创建 Action：将新照片同步到 Dropbox 的 `Dropbox/Camera Uploads/Android` 中。

设置 Action：

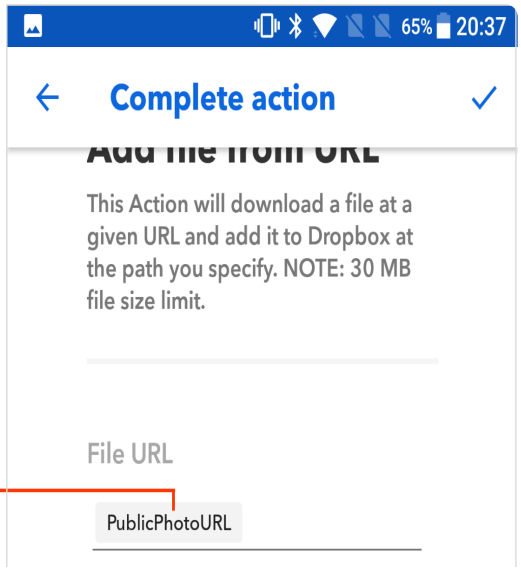
设置动作为「根据 URL 上传文件」

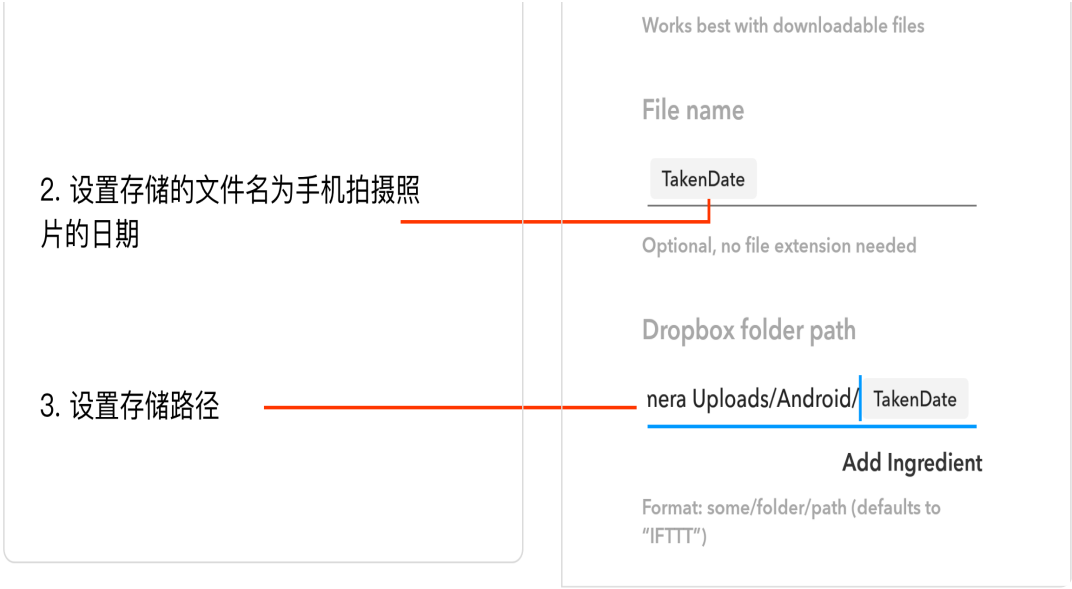

Dropbox lets people bring their documents, photos and videos everywhere and share them easily. Use Applets to sync your Dropbox uploads with other services, quickly add new files, and keep track of all your important photos, documents, and data – automatically.
Add file from URL
Create a text file
Append to a text file

设置 Action 1

设置 Action：

1. 设置存储的 URL 为手机照片的 URL （Android 自动提供）


65% 20:37
Complete action
This Action will download a file at a given URL and add it to Dropbox at the path you specify. NOTE: 30 MB file size limit.
File URL
PublicPhotoURL



设置 Action 2

Android 手机拍一张照片，由于系统支持，IFTTT 在后台就可以监控并上传照片到 Dropbox

1. 打开 Android 相机拍一张照片。
2. 由于系统支持，我们无须启动 IFTTT ，IFTTT 可以在后台监控到新照片，并上传到 Dropbox。
3. 打开 Dropbox ，查看新上传的照片。

第三部分：通过 Mac 上的 Hazel 应用，将 Android 拍摄的照片导入 iCloud

先梳理一下实现原理：

1. 安卓手机拍摄一张照片。
2. IFTTT 将照片上传到 Dropbox。

3. Dropbox 同步机制，将照片同步到 macOS 上的 `Dropbox/Camera Uploads/Android` 目录。
4. macOS 上的 Hazel 应用发现 `Dropbox/Camera Uploads/Android` 目录下的照片，根据规则将它导入到 macOS 的「照片」应用。
5. 由于照片应用使用的 iCloud 库，所以很快我们就能在 iPhone 上查看到 Android 手机拍摄的照片，做到所有照片都存储在 iCloud。

通过刚才「第二部分」的内容，我们借助 IFTTT + Dropbox 可以很方便的实现上边的步骤 1 和步骤 2，这里着重讲一下步骤 3 如何实现：

1. IFTTT 设置规则实现，每当发现有一张新的照片的时候，自动同步到 Dropbox 的 `Dropbox/Camera Uploads/Android` 目录下，具体实现参考「 第二部分：IFTTT 监控 Android 手机添加新照片，保存到 Dropbox」。
2. Dropbox 自动将照片同步到 macOS 的本地 `/Dropbox/Camera Uploads/Android` 目录，这一步利用 Dropbox 强大的同步机制。
3. 设置 Hazel 规则，发现 `/Dropbox/Camera Uploads/Android` 下有新的照片（证明有新的安卓照片被 IFTTT 导入到 Dropbox），自动将照片导入 macOS 的「照片」应用 。
4. 「照片」应用使用的是 iCloud 存储，所以照片会被同步到 iPhone 手机的「照片」应用中。

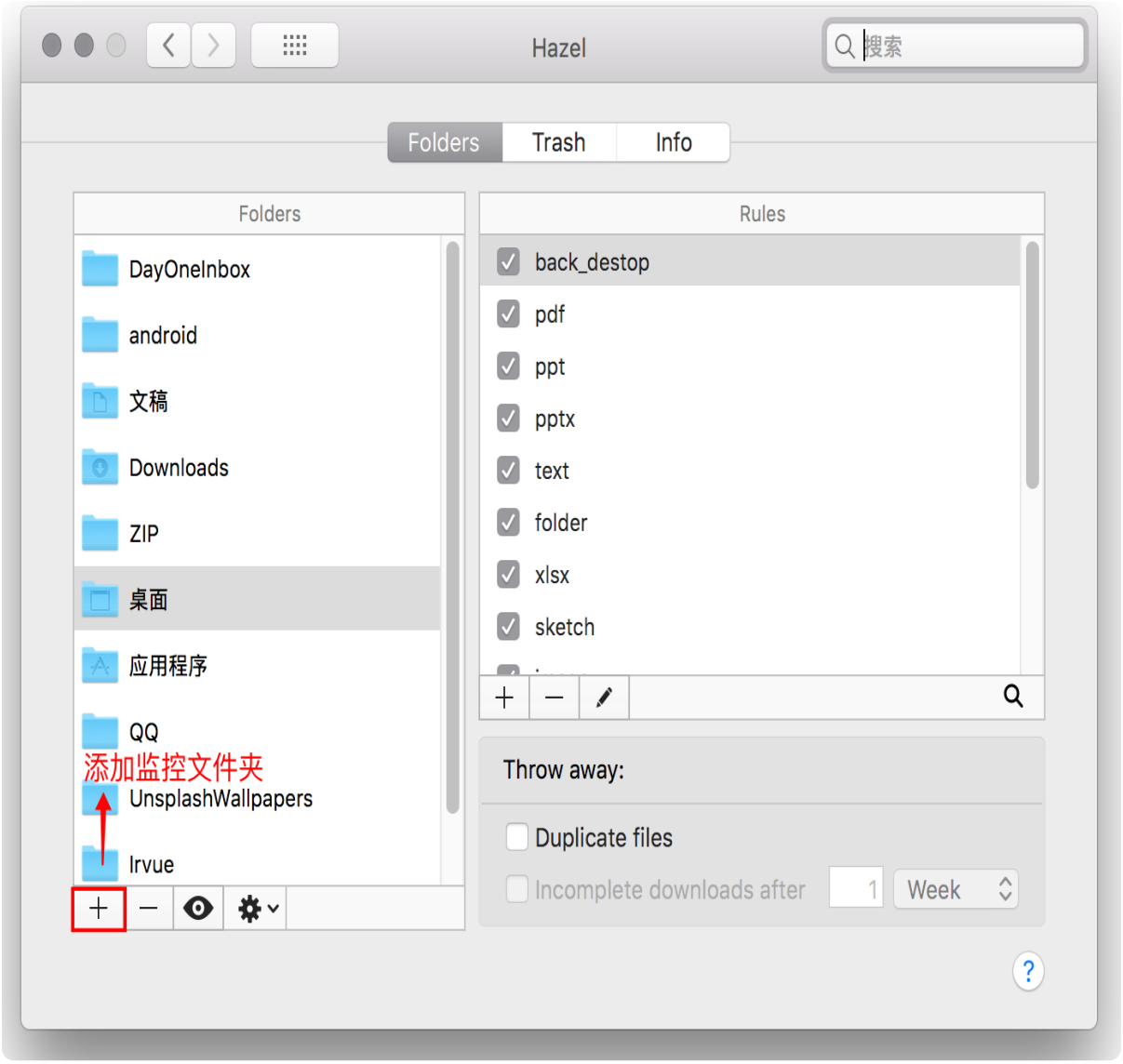
这里我们着重看一下 Hazel 如何实现的：

☞ 可以查看我之前的文章，了解 Hazel 使用《Hazel - 你的智能助手》

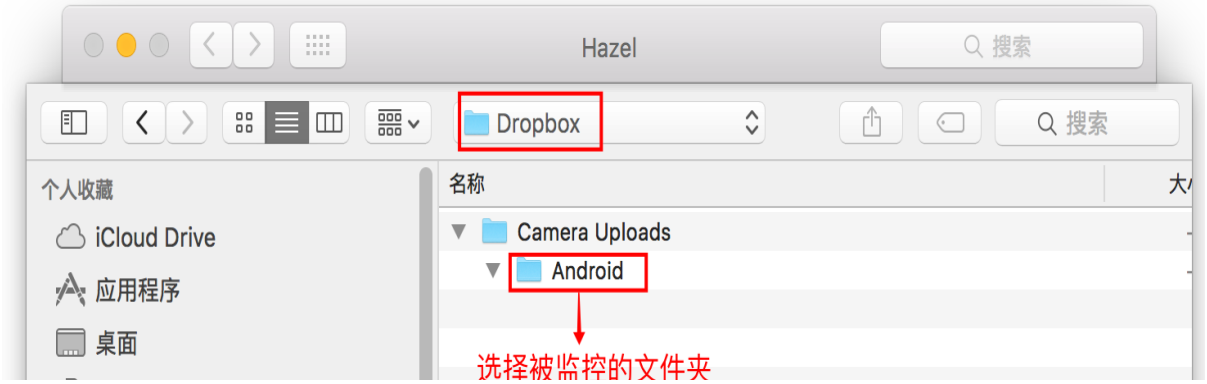
第一步，添加 Hazel 监控文件夹

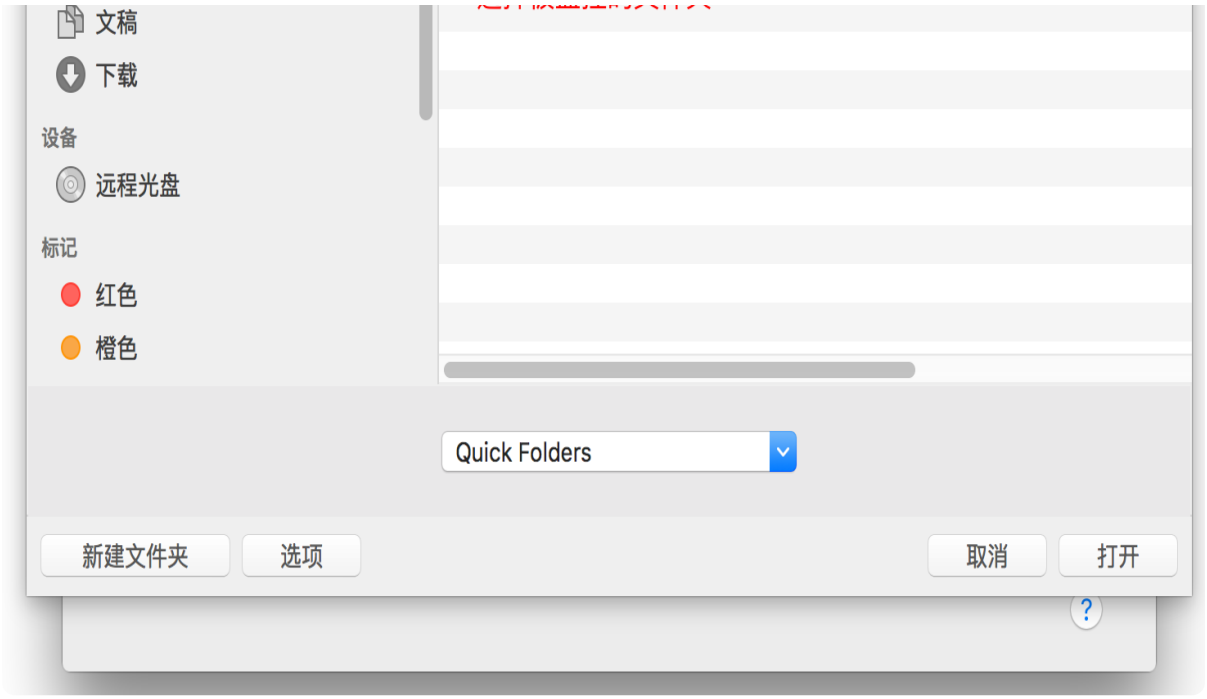
Hazel 软件通过设置监控目录，它会自动发现目录下文件，并且能够监控文件的新增、删除、创建时间、修改时间、等等条件，所以我们第一步将

`Dropbox/Camera Uploads/Android` 设置为监控目录。

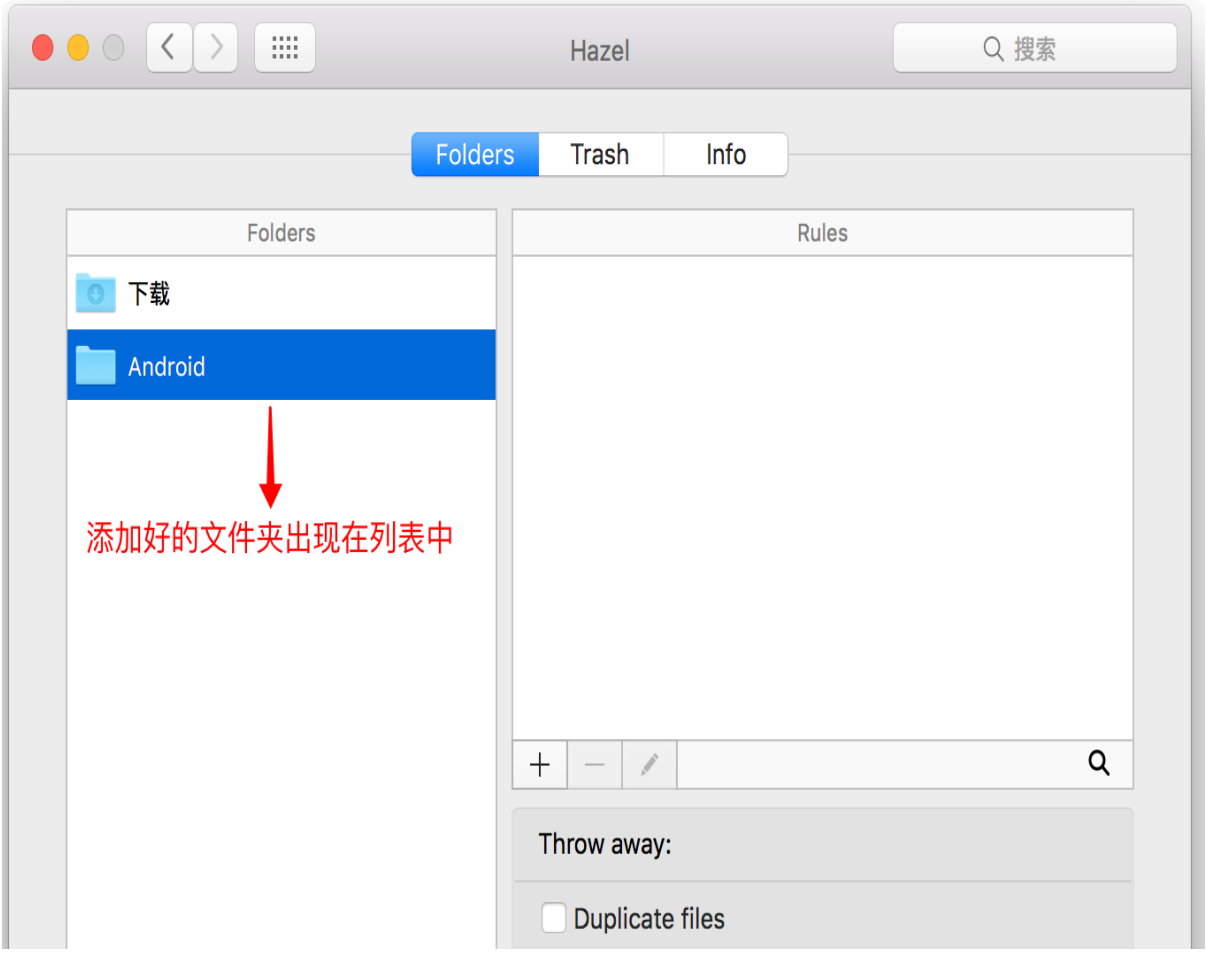


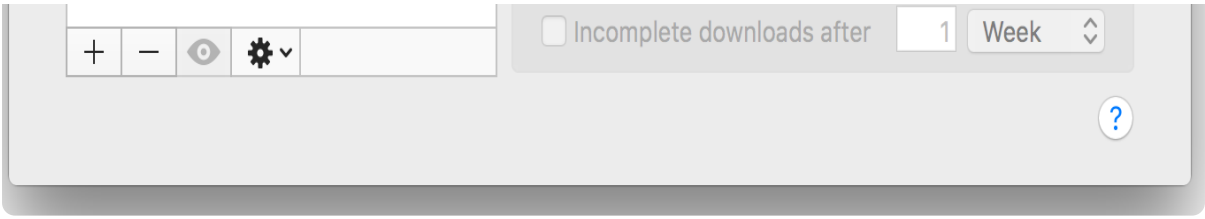
添加监控目录




文稿
下载
设备
远程光盘
标记
红色
橙色
Quick Folders
新建文件夹
选项
取消
打开

选择监控文件夹


Hazel
搜索
Folders
Trash
Info
Folders
Rules
下载
Android
添加好的文件夹出现在列表中
Throw away:
Duplicate files

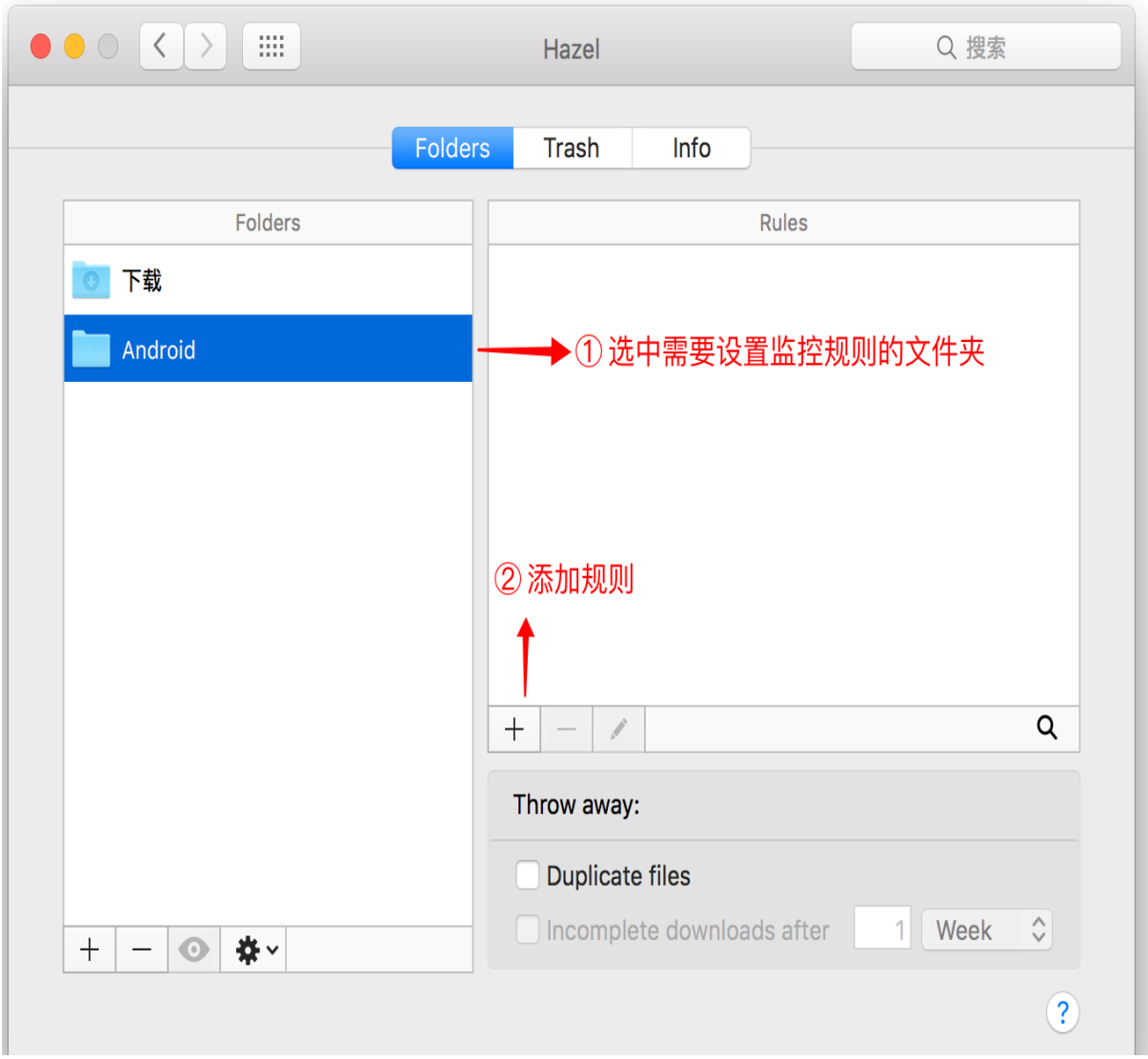


添加好的目录

第二步，添加规则，发现新的照片导入到 iCloud

接下来就是 Hazel 最核心的功能，监控新增文件，并执行规则，大家继续跟着我一步步创建规则实现我们的需求。

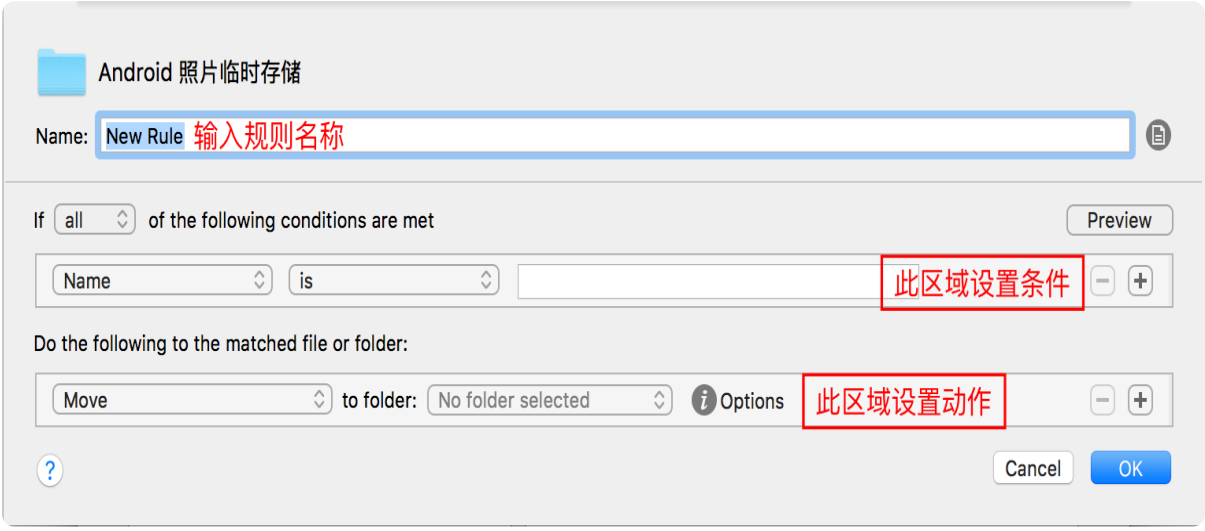
选中监控文件夹，点击 + 添加规则：



添加规则

添加规则：

认识一下规则设置界面：



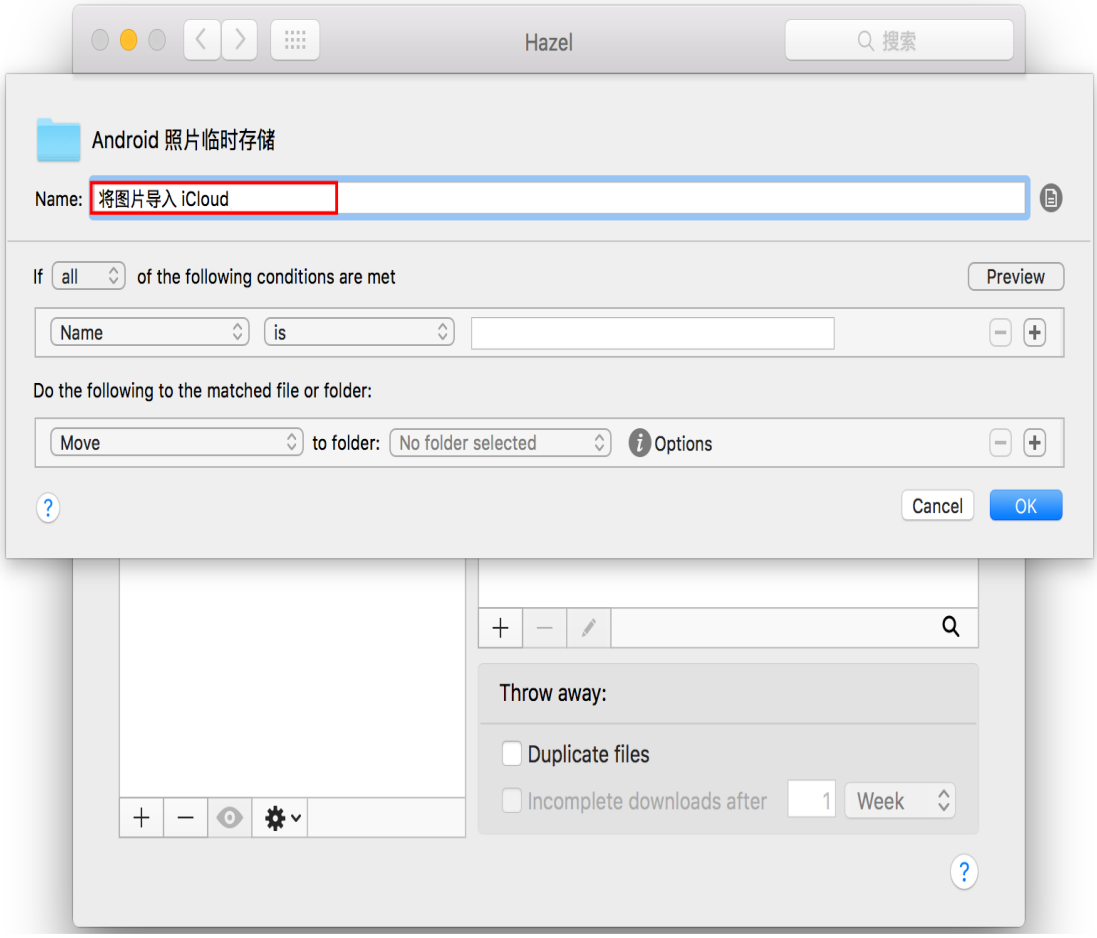
设置规则页面

大家看到整个界面被分为三个区域：

- 区域一：规则名称
- 区域二：设置条件区域
- 区域三：设置动作区域

这里很像 IFTTT，同样是「如果 A 条件被满足，那么 B 动作被执行」，回到我们的需求，我们的需求是「如果文件夹里新增一张照片，那么就将照片导入到 iCloud」。

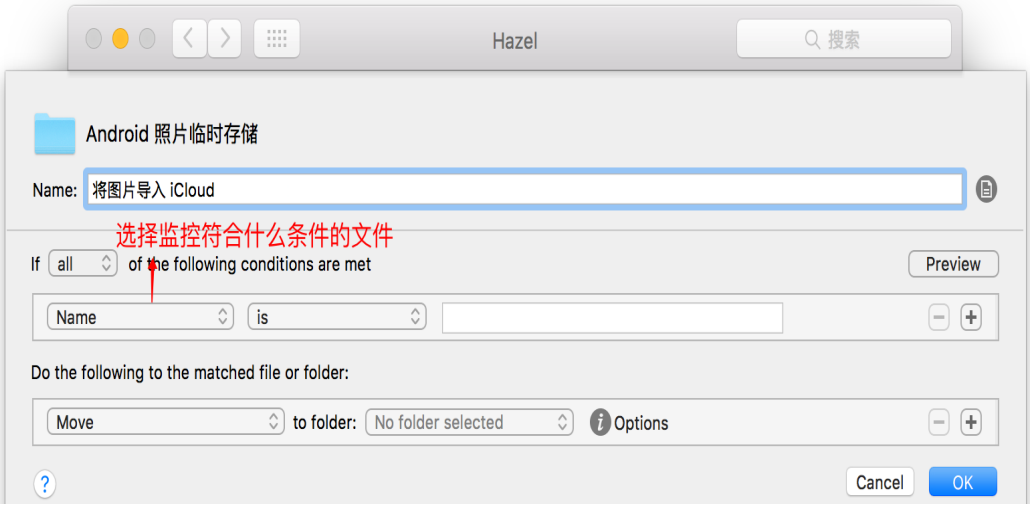
1. 设置规则名称：

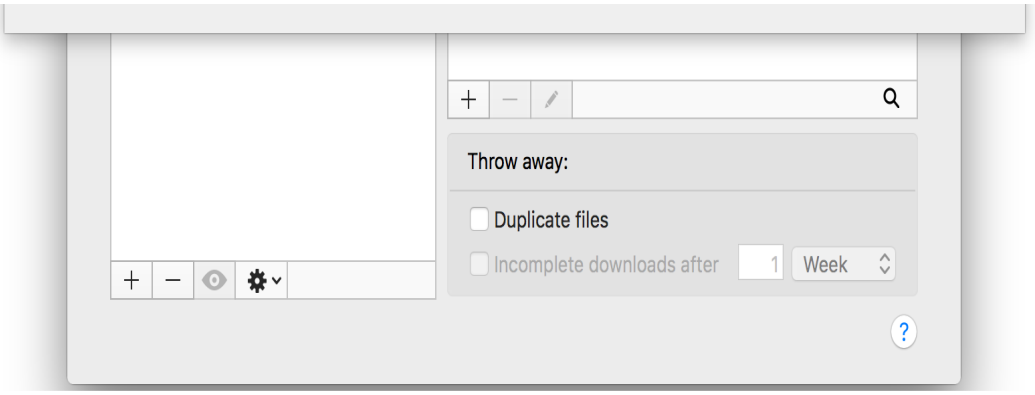


设置规则名称

2. 设置条件，监控文件夹下所有图片文件：

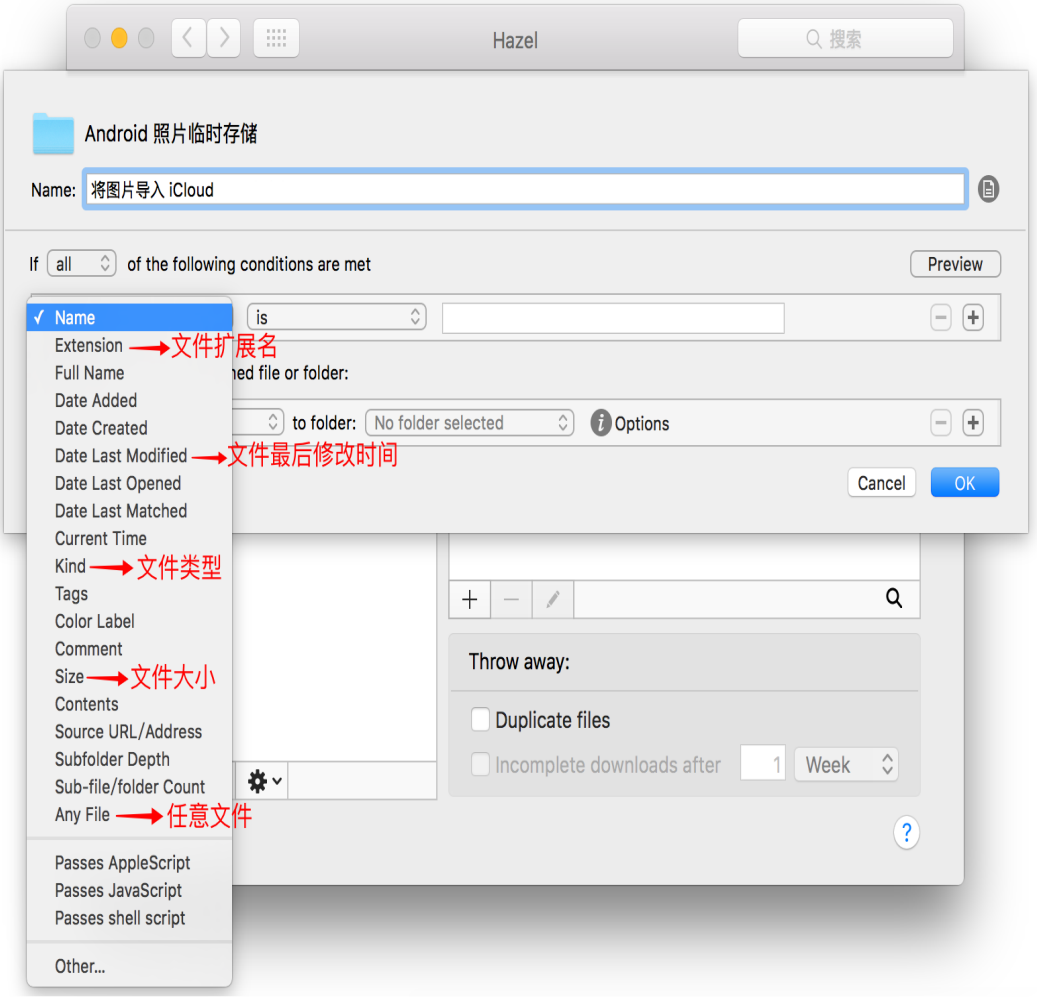
    1. 选择条件：





选择规则条件

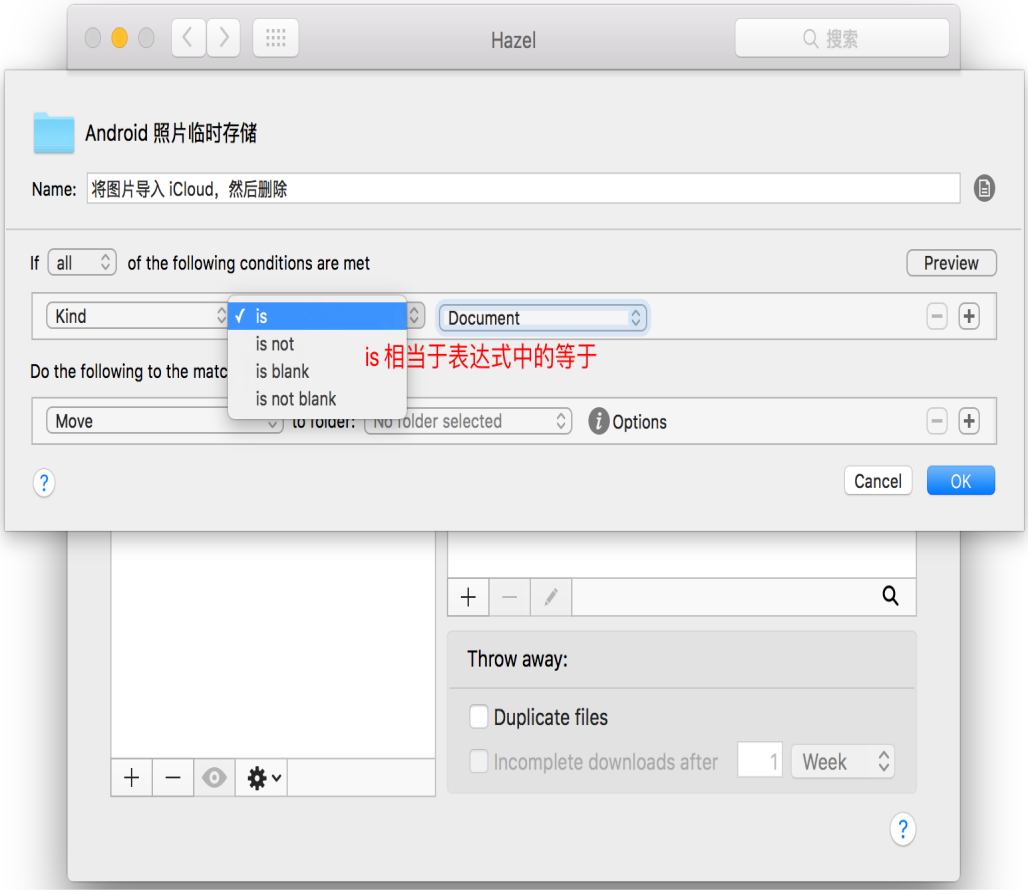
2. 查看所有支持的条件：



规则条件说明

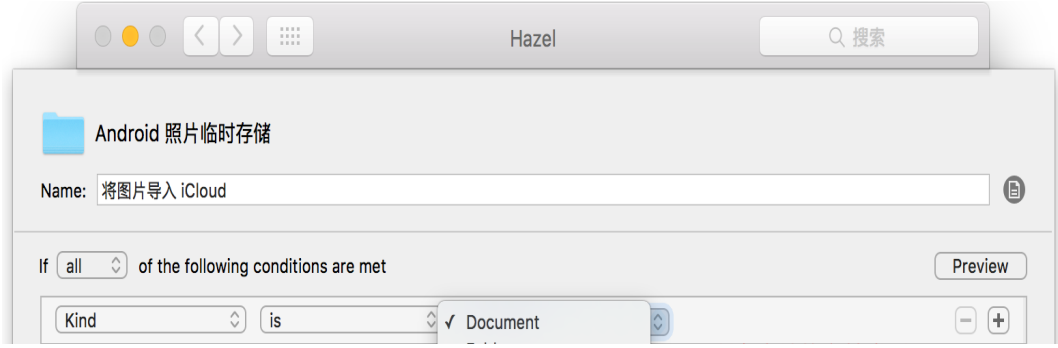
很显然我们选择 **Kind**（文件类型）这个条件即可，因为我们的条件是所有图片类型的文件进行下一步处理。

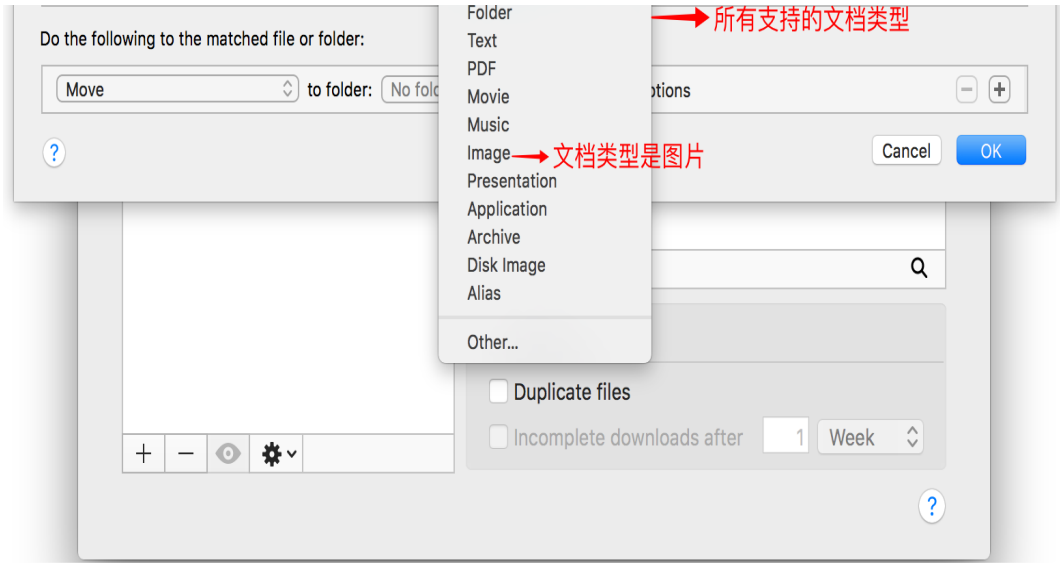
3. 选择关键字「IS」，代表「文件类型是」后边的选择值：



选择规则条件

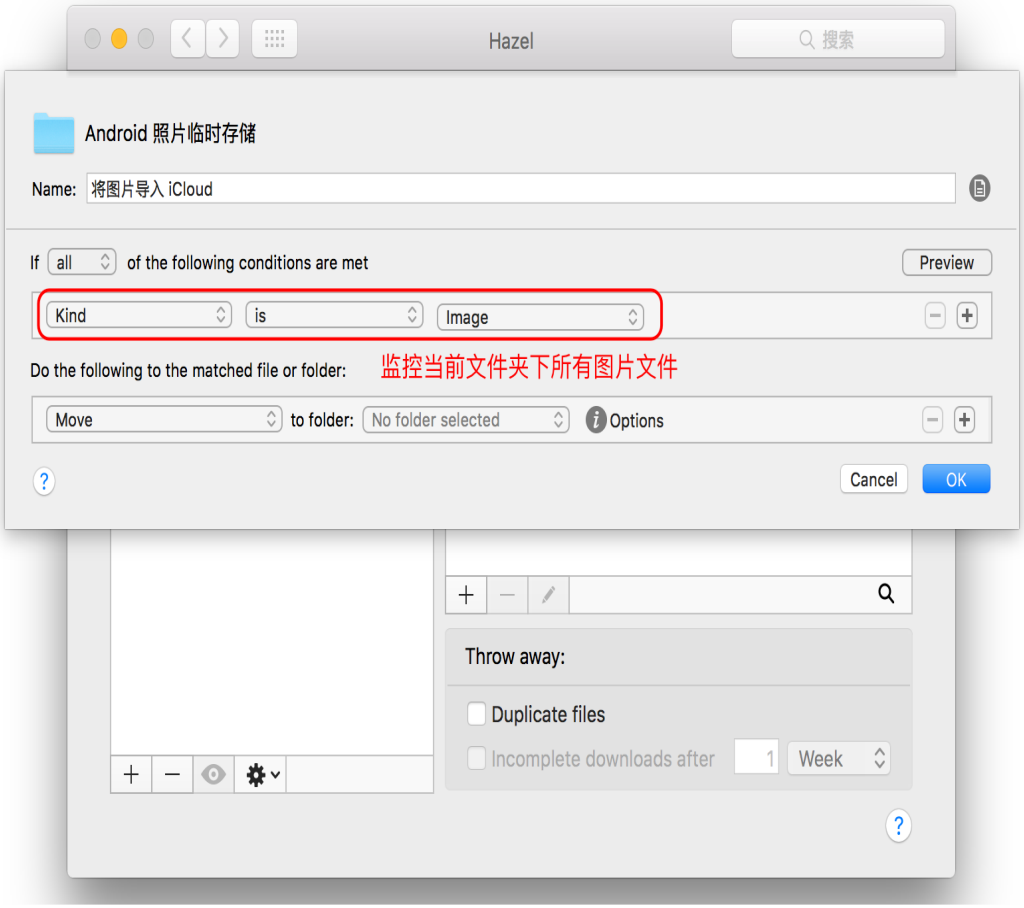
4. 选择类型为 image：





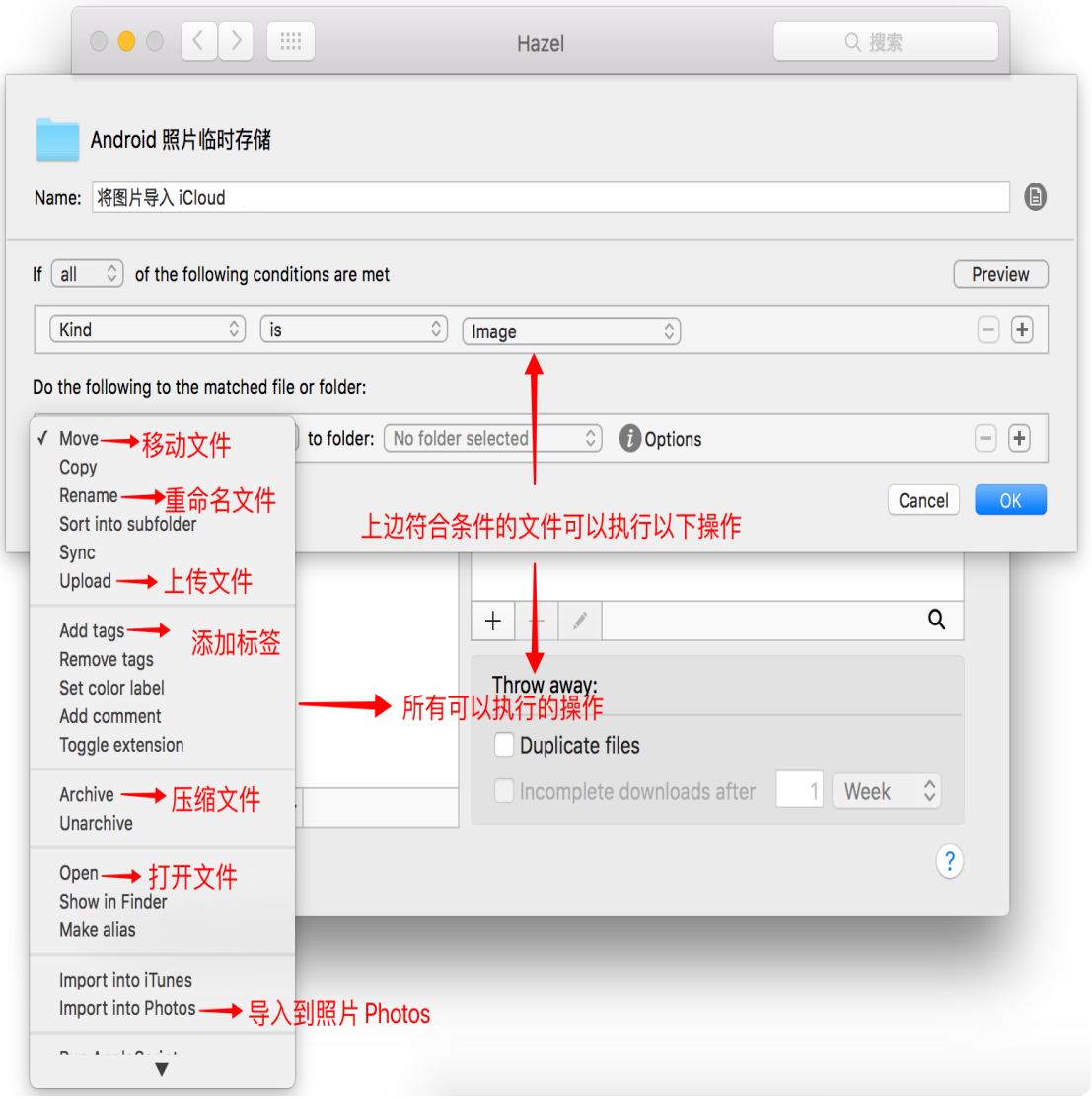
规则条件说明

5. 最终设置好的条件：



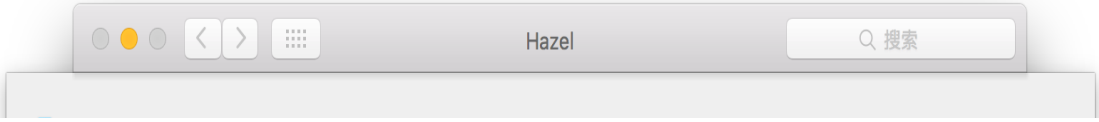
设置条件最终效果

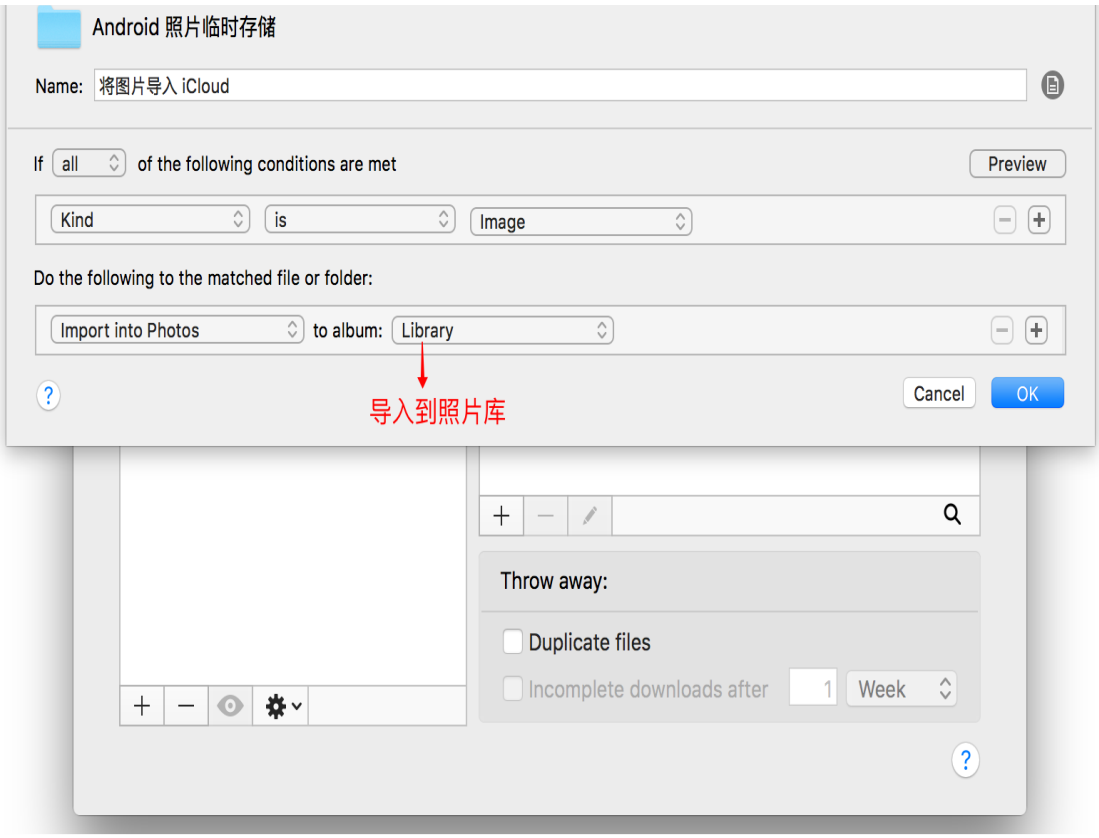
3. 设置动作，将图片导入到 iCloud，选择动作类型



选择动作类型

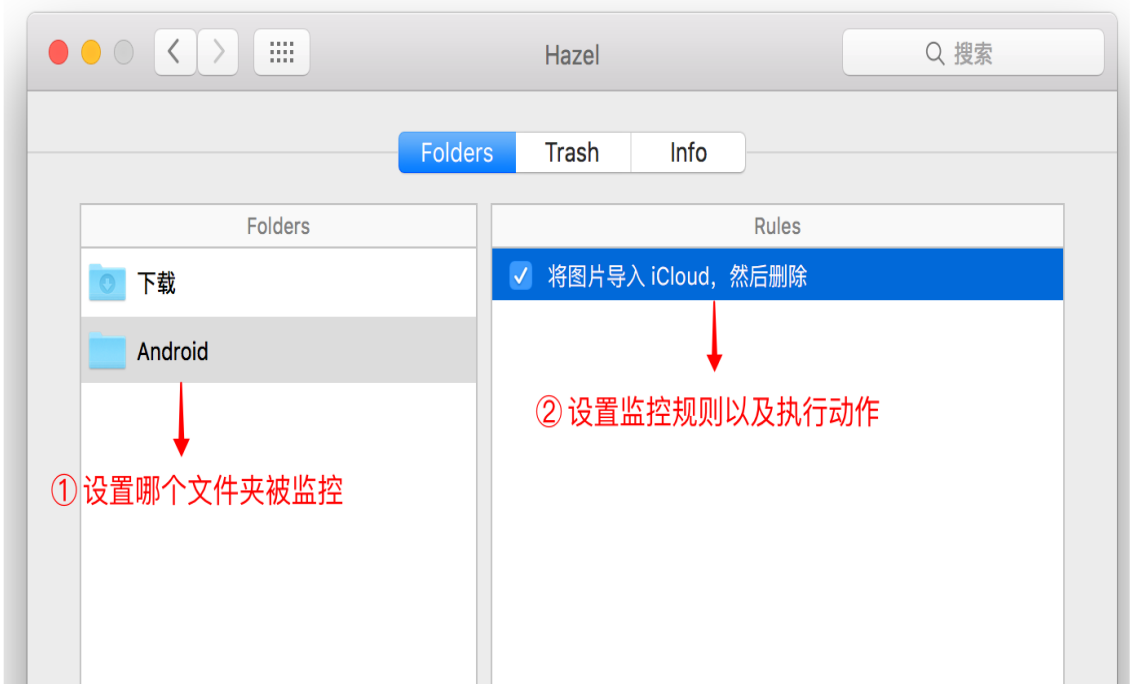
我们的需求是将将照片同步到 iCloud，很显然选择 **Import into Photos**，导入到照片应用。

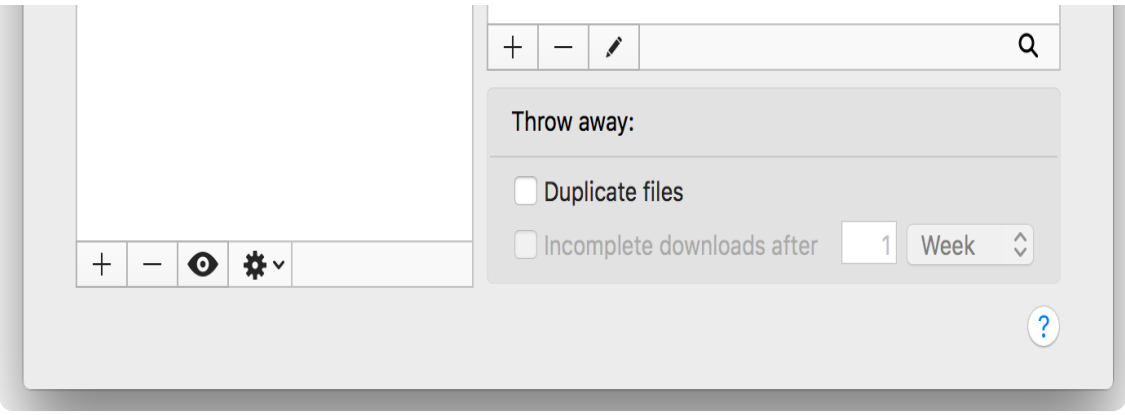




设置动作内容

4. 保存规则





保存规则

设置好 Hazel 的规则，它会自动监控 `/Dropbox/Camera Uploads/Android` 文件夹下，如果有新增的图片，执行规则将图片导入到 Photos（照片应用）。这样就完成了 Android 手机拍照自动同步到 iCloud。

小结以及关于自动保存照片的探讨

还记得我们自动存储照片的目标吗：

1. 我希望随时都随地打开 iPhone 能够看到我在 iPhone 和 Android 手机上拍的所有照片；
2. 我希望我所有的照片在 iCloud 和 Dropbox 都有备份。

为了实现目标我们通过三部分完成：

第一部分：IFTTT 监控 iPhone 添加新照片，保存到 Dropbox

1. 第一步，创建自动化任务，实现 iOS 照片自动备份到 Dropbox。
2. 第二步，iOS 拍一张照片，打开 IFTTT App 使得条件被动触发。

第二部分：IFTTT 监控 Android 手机添加新照片，保存到 Dropbox

1. 第一步，创建自动化任务，实现 Android 手机拍照自动备份到 Dropbox。
2. 第二步，Android 手机拍一张照片，由于系统支持，IFTTT 在后台就可以监控并上传照片到 Dropbox。

第三部分：通过 Mac 上的 Hazel 应用，将 Android 拍摄的照片导入 iCloud

1. 第一步，添加 Hazel 监控文件夹。
2. 第二步，添加规则，发现新的照片导入到 iCloud

这个流程总体上看比较复杂，涉及内容比较多，希望大家动手操作，如果你对流程有优化意见，也建议大家提出来共同讨论。

IFTTT 凭借自身的强大机制，通过设置规则可以让照片同步到 Dropbox，但是由于 iOS 系统的限制，IFTTT 不能像 Android 手机那样「后台自动上传」，我们需要定期打开 IFTTT，让它做一次同步。另外，Dropbox 免费的空间最多 22 GB（大约），如果照片太多也会遇到「空间不足」等问题。那么有没有简单的替代方案呢?

替代方案 1：Dropbox 相机上传

iOS 9 以后系统开放了「后台刷新机制」，Dropbox 客户端通过「后台刷新机制」可以做到「静默状态」下监控手机位置变化，定期进行照片上传操作。这个功能比 IFTTT 更加方便一些（当然也更耗电），感兴趣的同学可以尝试使用 Dropbox 客户端的「相机上传」功能替代「第一部分：IFTTT 监控 iPhone 添加新照片，保存到 Dropbox」。

Dropbox 相机上传的优点：

1. 后台静默上传，不用总是启动。
2. 可以设置只在 Wi-Fi 下上传，节省流量。

Dropbox 相机上传的缺点：比 IFTTT 更耗电。

替代方案 2：Google 照片同步

Google 照片，Google 出品的照片管理应用，具有非常优秀的面部识别和智能相册功能。凡是用过的的用户，经常会被 Google 的智能整理惊倒，如果你有超过 5 年的照片，推荐你体验一下 Google 的黑科技，相信一定会有不凡体验。

Google 照片最诱人的是我们可以拥有无限的上传空间，前提是上传大小设置成「上传高品质（文件缩小，画质不减）」。相比之下 Dropbox 免费最多使用 22 GB（大约） 空间就捉襟见肘了。

Google 照片的优点：

1. 近乎无限的存储空间，从此不用担心空间不足。
2. 智能相册，面部识别，经常给你带来意外惊喜。

Google 照片的缺点：

1. 无法做到后台静默上上传。
2. 照片不能本地同步，只能在 App 或者网页中查看。

05. 总结

本篇是带你玩转 IFTTT 系列教程的第三篇**本地应用篇**，在本篇里我们分别了解了 iOS Reminders、Android SMS、iOS Calendar、iOS Photos、Android Photos、button Widget、Note Widget、Camera Widget、iOS Health 等本地服务，利用它们可以让本地应用自动化，为我们节约时间，提高生产力。

如果大家觉得教程提供的事例太少，可以尝试搜索官方提供的事例进行研究：

- 官方提供的它们最喜欢的自动化实例，点击访问
- 官方制作的自动化实例合集，点击访问
- 官网 或 app 提供的**探索**功能，点击访问
- 官网 或者 app 提供的**搜索**功能，可按类型筛选，点击访问

接下来的文章里，我将陆续介绍 IFTTT 和智能硬件相结合的一些玩法，欢迎大家持续关注本系列教程。

---

1. Widget，iOS 通知中心小插件，可以显示日历、任务等列表信息，也可以通过几个按

钮快速实现某些功能，甚至还可以玩小游戏。 ↩

2. API（Application Programming Interface 应用服务接口）是一些预先定义的网络服务接口，应用通过 API 实现对服务的操作。如果一个服务的 API 完全开放的话，那么任何其他应用或者服务都可以通过 API 来享受这个服务提供的功能。 ↩
3. Google Sheets，Google 出品的云端数据表，提供了非常优秀的 API，方便我们在云端处理、统计和计算。 ↩
4. 3：存储 3 份完整文件，一份原件加上两份拷贝。2：将文件起码保持在两种不同的介质上。1：将一份拷贝保存在异地。 ↩
5. Hazel，就是你的智能文档管理助手。简单说，它可以根据制定的规则，自动执行命令或处理业务。借助其强大的脚本和系统整合能力，给你的无限可能，不折不扣的一款神器。 ↩

上一期
答疑汇总 02

下一期
第四章 | IFTTT 的进阶使用